Journal of Computer and System Sciences 76 (2010) 561–578

Contents lists available at ScienceDirect

Journal of Computer and System Sciences

www.elsevier.com/locate/jcss

Generalized modal satisfiability ☆

Edith Hemaspaandra [a], Henning Schnoor [b,*], Ilka Schnoor [c]

[a] Department of Computer Science, Rochester Institute of Technology, Rochester, NY 14623, USA
[b] Institut für Informatik, Christian-Albrechts-Universität Kiel, Christian-Albrechts-Platz 4, D-24098 Kiel, Germany
[c] Institut für Theoretische Informatik, Universität Lübeck, Ratzeburger Allee 160, D-23538 Lübeck, Germany

ARTICLE INFO

Article history:
Received 3 April 2008
Received in revised form 5 October 2009
Available online 17 October 2009

Keywords:
Computational complexity
Modal logic

ABSTRACT

It is well known that modal satisfiability is PSPACE-complete (Ladner (1977) [21]). However, the complexity may decrease if we restrict the set of propositional operators used. Note that there exist an infinite number of propositional operators, since a propositional operator is simply a Boolean function. We completely classify the complexity of modal satisfiability for every finite set of propositional operators, i.e., in contrast to previous work, we classify an infinite number of problems. We show that, depending on the set of propositional operators, modal satisfiability is PSPACE-complete, coNP-complete, or in P. We obtain this trichotomy not only for modal formulas, but also for their more succinct representation using modal circuits. We consider both the uni-modal and the multi-modal cases, and study the dual problem of validity as well.

1. Introduction

Modal logics are valuable tools in computer science, since they are often a good compromise between expressiveness and decidability. Standard applications of modal logics are in artificial intelligence [27,26], and cryptographic and other protocols [12,8,15,22]. More recent applications include a new modal language called Versatile Event Logic [4], and the usage to characterize the relationship among belief, information acquisition, and trust [25].

Applications of modal logic for solving practical problems obviously require a study of the computational complexity of various aspects of modal logics. A central computational problem associated with any logic is the satisfiability problem, that is to decide whether a given formula has a model. The first complexity results for the modal satisfiability problem were achieved by Ladner [21]. He showed that the basic modal satisfiability problem is PSPACE-complete. There is a rich literature on the complexity of variants of the modal satisfiability problem, important works include the paper by Halpern and Moses [14] on multi-modal logics. Recently, PSPACE-algorithms for a wide class of modal logics were presented by Schröder and Pattinson [36].

For modal logics to be used in practice, a lower complexity of the satisfiability problem than the aforementioned PSPACE-hardness is desirable. It turns out that for many applications, the full power of modal logic is not necessary. There are various ways of defining restrictions of modal logics which potentially lead to a computationally easier version of the satisfiability problem that have been studied: Variations of modal logics are achieved by restricting the class of considered models, e.g., instead of allowing arbitrary graphs, classical examples of logics only allow reflexive, transitive, or symmetric graphs

☆ Supported in part by the DAAD Postdoc Program, by grants NSF-CCR-0311021, NSF-IIS-0713061, and DFG VO 630/5-1, and by a Friedrich Wilhelm Bessel Research Award. Work done in part while the second and third authors were at the Leibniz Universität Hannover and at the Rochester Institute of Technology. An earlier version of some of the results appeared as Bauland et al. (2006) [1].

* Corresponding author.
E-mail addresses: eh@cs.rit.edu (E. Hemaspaandra), schnoor@ti.informatik.uni-kiel.de (H. Schnoor), schnoor@tcs.uni-luebeck.de (I. Schnoor).

0022-0000/$ – see front matter
doi:10.1016/j.jcss.2009.10.011

as models. Many complexity results for logics defined in this way have been achieved: Initial results for many important classes are present in the above-mentioned work by Ladner [21]. Recently, Hemaspaandra and Schnoor considered a uniform generalization of many of these examples [19]. It should be noted that such restrictions do not necessarily decrease the complexity; for many common restrictions, the complexity remains the same [21,14] and it is even possible that the complexity increases. In [16], Hemaspaandra showed that the complexity of the global satisfiability problem increases from EXPTIME-complete to undecidable by restricting the graphs to those in which every node has at least two successors and at most three 2-step successors.

Another way of restricting modal logics is to change the syntax rather than the semantics, i.e., restrict the structure of the considered modal formulas. Syntactical restrictions are known to naturally reduce the complexity of many decision problems in logic. In propositional logic, well-known examples are the satisfiability problems for Horn formulas, 2CNF formulas, or formulas describing monotone functions: All of these can be solved in polynomial time, while the general propositional satisfiability problem is NP-complete. Syntactical restrictions have been considered in the context of modal logics before: Halpern showed that the complexity of the modal satisfiability problem decreases to linear time when restricting the number of variables and nesting degree of modal operators [13]. Restricted modal languages where only a subset of the relevant modal operators are allowed have been studied in the context of linear temporal logic [37]. Some description logics can be viewed as modal logic with a restriction on the propositional operators that are allowed. For the complexity of description logics, see, e.g., [34,10,11]. For the complexity of modal logic with other restrictions on the set of operators, see [17].

The approach we take in the present paper is to generalize the occurring propositional operators in the formulas. Instead of the operators $\wedge, \vee$ and negation, we allow the appearing operators to represent arbitrary Boolean functions. In particular, there are an infinite number of Boolean operators. We completely classify the complexity of modal satisfiability for every finite set of propositional operators. The restriction on the propositional operators leads to a classification following the structure of Post's Lattice [29], a tool that has been applied in similar contexts before: For propositional logic, Lewis showed that the satisfiability problem is dichotomic: Depending on the set of operators, propositional satisfiability is either NP-complete or solvable in polynomial time [24]. For modal satisfiability, we achieve a trichotomy: For the modal logic K, the satisfiability problem is PSPACE-complete, coNP-complete, or in P. We also achieve a full classification for the logic KD (in this case, we show a PSPACE/P-dichotomy), and almost complete classifications for the logics T, S4, and S5.

When considering sets of operations which do not include negation, the complexity for the cases where one modal operator is allowed sometimes differs from the case where we allow both operator $\Diamond$ and its dual operator $\Box$. With only one of these, modal satisfiability is PSPACE-complete exactly in those cases in which propositional satisfiability is NP-complete. When we allow both modal operators, the jump to PSPACE-completeness happens earlier, i.e., with a set of operations with less expressive power.

We consider several generalizations of the problems outlined above. In particular, we introduce *modal circuits* as a succinct way of representing modal formulas. We show that this does not give us a significantly different complexity than the formula case. We also consider multi-modal logics, in which several independent modal operators are introduced.

In addition to the satisfiability problem, we also study the validity (tautology) problem, where we do not ask whether a formula is satisfiable, but whether it is true in every possible model. Since our restricted modal languages do not always include negation, the complexity of this problem turns out to be different from, but related to, the complexity of the satisfiability problem.

An interesting case in our classifications is the case where we only allow the propositional exclusive-or and constants as propositional operators. For purely propositional logics, it is very easy to see that satisfiability for these formulas (essentially linear equations over GF(2)) can be decided in polynomial time. In the case of modal logics, an analogous result holds, but the proof requires significantly more work.

The structure of the paper is as follows: In Section 2, we introduce the necessary definitions, recall results from the literature, and prove some basic facts about our problems. Section 3 contains our main results: The complete classification of the complexity of the modal satisfiability problem for every possible set of Boolean operators. In Section 4 we prove a relationship between satisfiability and validity implying a full classification of this problem as well. We conclude in Section 5 with some open questions for future research.

2. Preliminaries

2.1. Modal logic

Modal logic is an extension of classical propositional logic that talks about "possible worlds." We first informally explain the usual uni-modal logic, and then formally introduce the multi-modal case. Uni-modal logics enrich the vocabulary of propositional logic with an additional unary modal operator $\Diamond$. A model for a given formula consists of a directed graph with propositional assignments. To be more precise, a *frame* consists of a set W of "worlds," and a "successor" relation $R \subseteq W \times W$. For $(w, w') \in R$, we say w' is a *successor* of w. A *model* M consists of a frame (W, R), a set X of propositional variables, and a function $\pi : X \to \mathcal{P}(W)$. The intuition is that for $x \in X$, $\pi(x)$ denotes the set of worlds in which the variable x is true. The operator $\Box$ is the dual operator to $\Diamond$, $\Box\varphi$ is defined as $\neg\Diamond\neg\varphi$. Intuitively, $\Diamond\varphi$ means "there is a successor world in which φ holds," and $\Box\varphi$ means "φ holds in all successor worlds." For a class $\mathcal{F}$ of frames, we say that a model M is an $\mathcal{F}$-model if the underlying frame is an element of $\mathcal{F}$.

Table 1
Classes of frames.

K	All frames
KD	Frames in which every world has a successor
K4	Transitive frames
S4	Frames that are reflexive and transitive
S5	Frames that are reflexive, transitive, and symmetric
T	Reflexive frames

In multi-modal logic, a finite number of these modal operators is considered, where each operator $\Diamond_i$ corresponds to an individual successor relation R_i. For a modal logic with k modalities, a frame again consists of a set W of worlds, and successor relations $R_1, \ldots, R_k \subseteq W \times W$. If $(w, w') \in R_i$, we say that w' is an *i-successor* of w. For a formula φ built over propositional variables, propositional operators $\wedge$ and $\neg$, and the unary modal operators $\Diamond_1, \ldots, \Diamond_k$, we define what "$\varphi$ holds at world w" means for a model M (or M, w *satisfies* φ) with assignment function π, written as $M, w \models \varphi$.

- If φ is a propositional variable x, then $M, w \models \varphi$ if and only if $w \in \pi(x)$,
- $M, w \models \varphi_1 \wedge \varphi_2$ if and only if ($M, w \models \varphi_1$ and $M, w \models \varphi_2$),
- $M, w \models \neg\varphi$ if and only if $M, w \not\models \varphi$,
- for $i \in \{1, \ldots, k\}$, $M, w \models \Diamond_i\varphi$ if and only if there is a world $w' \in W$ such that $(w, w') \in R_i$ and $M, w' \models \varphi$.

Analogously to the unimodal case, the operator $\Box_i$ is defined as $\Box_i\varphi = \neg\Diamond_i\neg\varphi$. For a class $\mathcal{F}$ of frames, we say a formula φ is $\mathcal{F}$-satisfiable if there exists an $\mathcal{F}$-model $M = (W, R, \pi)$ and a world $w \in W$ such that $M, w \models \varphi$. For modal formulas φ and ψ, we write $\varphi \equiv_{\mathcal{F}} \psi$ if for every world in every $\mathcal{F}$-model, φ holds if and only if ψ holds. Note that a formula φ is $\mathcal{F}$-satisfiable iff $\varphi \not\equiv_{\mathcal{F}} 0$. Similarly, we say that φ is an $\mathcal{F}$-tautology if $\varphi \equiv_{\mathcal{F}} 1$, and finally φ is $\mathcal{F}$-constant if $\varphi \equiv_{\mathcal{F}} 0$ or $\varphi \equiv_{\mathcal{F}} 1$.

We now define the classes of frames that are most commonly used in applications of modal logic (see Table 1). To see how these frames correspond to axioms and proof systems, see, for example, [5, Section 4.3]. Again, we first consider the uni-modal case and then present the natural generalizations to multi-modal logics. K is the class of all frames, KD is the class of frames in which every world has a successor, i.e., for all $w \in W$, there is a $w' \in W$ such that $(w, w') \in R$. T is the class of reflexive frames, K4 is the class of transitive frames, S4 is the class of frames that are both reflexive and transitive, and S5 is the class of reflexive, symmetric, and transitive frames. The *reflexive singleton* is the frame consisting of one world w, and the relation $\{(w, w)\}$. Note that all classes of frames $\mathcal{F}$ described above contain the reflexive singleton. Similarly, the *irreflexive singleton* is the frame consisting of one world, and an empty successor relation.

For multi-modal logics, the generalizations are obvious: For a class of frames $\mathcal{F}$ as previously defined, we say that the class $\mathcal{F}_k$ contains those frames $(W, R_1, \ldots, R_k)$, where $(W, R_i) \in \mathcal{F}$ for all $i \in \{1, \ldots, k\}$. In particular, a multi-modal reflexive singleton consists of the set of worlds $W = \{w\}$ where each successor relation consists of the pair (w, w), and the multi-modal irreflexive singleton consists of the same set of worlds where all of the successor relations are empty. If the number k of modal operators is clear from the context, we often simply write $\mathcal{F}$ instead of $\mathcal{F}_k$, speak about the reflexive singleton, etc.

2.2. Generalized formulas and circuits

We now consider a more general notion of modal formulas, whose propositional analog has been studied extensively. We generalize the notion of a modal formula in two ways: First, instead of allowing the usual propositional operators $\wedge, \vee$, and $\neg$, we allow operators defined by arbitrary Boolean functions. Second, we study circuits as succinct representations of formulas. Intuitively, a circuit is a generalization of a formula in the same way as a directed acyclic graph is a generalization of a tree, since formulas directly correspond to tree-like circuits. To be more precise, for a finite set B of Boolean functions, a *modal B-circuit* is a generalization of a propositional Boolean circuit (see, e.g., [38] for an introduction to Boolean circuits) with gates for functions from B and additional gates representing the modal operators $\Diamond_i$ or $\Box_i$. Boolean circuits are a standard way to succinctly represent Boolean functions. Formally, we define the following:

Definition 2.1. Let B be a finite set of Boolean functions, let $M \subseteq \{\Box, \Diamond\}$, and let $k \geqslant 0$. A circuit in $\mathrm{MCIRC}^k_M(B)$ is a tuple $C = (V, E, \alpha, \beta, out)$ where (V, E) is a finite directed acyclic graph, $\alpha : E \to \mathbb{N}$ is an injective function, β is a function that assigns to each element from V a function from B, one of the modal operators $\Box_1, \ldots, \Box_k, \Diamond_1, \ldots, \Diamond_k$, or a propositional variable, and $out \in V$, such that:

- If $v \in V$ has in-degree 0, then $\beta(v)$ is a propositional variable or $\beta(v)$ is a 0-ary function (a constant) from B.
- If $v \in V$ has in-degree 1, then $\beta(v)$ is a unary function from B or, for some $i \in \{1, \ldots, k\}$, one of the operators $\Box_i$ (if $\Box \in M$) or $\Diamond_i$ (if $\Diamond \in M$).
- If $v \in V$ has in-degree $d > 1$, then $\beta(v)$ is a d-ary function from B.

By definition, $\mathrm{MCIRC}^k_M(B)$ contains the modal circuits that use the following as operators: functions from B, the modal operators $\Diamond_1, \ldots, \Diamond_k$ if $\Diamond \in M$, and $\Box_1, \ldots, \Box_k$ if $\Box \in M$. Nodes $v \in V$ are called *gates* of C, $\beta(v)$ is the *gate-type* of v. The node *out* is the *output-gate* of C. The function α is needed to define the order of arguments for non-symmetric functions, as will become apparent shortly. The *size* of a modal circuit C is the number of gates: $|C| := |V|$.

In addition to circuits, we also study the usually considered case of modal formulas. A *modal B-formula* is a modal B-circuit where each gate has out-degree $\leqslant 1$. This corresponds to the standard notion of a formula: Such a circuit can be written down as a formula, e.g., in prefix notation, without growing significantly in size. Semantically we interpret a circuit as a succinct representation of its formula expansion. For a modal B-circuit C, the *modal depth of C*, $md(C)$, is the maximal number of gates representing modal operators on a directed path in the graph. If there are no modal gates (i.e., gates $v \in C$ such that $\beta(v) \in \{\Box_i, \Diamond_i\}$ for some i) then φ_C is a *propositional Boolean formula* and C is a *propositional Boolean circuit*.

In order to define the semantics of the circuits defined above, we relate them to formulas in the following natural way: The circuit C represents the modal formula φ_C that is inductively defined by a modal B-formula φ_v for every gate v in C:

Definition 2.2. Let B be a finite set of Boolean functions, let $M \subseteq \{\Box, \Diamond\}$, let $k \in \mathbb{N}$, and let $C \in \mathrm{MCIRC}^k_M(B)$, let V be the set of gates of C.

- If $v \in V$ has in-degree 0, then $\varphi_v := \beta(v)$.
- Let $v \in V$ have in-degree $l > 0$, and let $v_1, \ldots, v_l$ be the predecessor gates of v such that $\alpha((v_1, v)) < \cdots < \alpha((v_l, v))$. Then let $\varphi_v := \beta(v)(\varphi_{v_1}, \ldots, \varphi_{v_l})$.
- Finally, we define φ_C as φ_{out}. We call φ_C the *formula expansion of C*.

Since every Boolean function can be expressed using only conjunction and negation, the semantics for circuits allowing arbitrary Boolean functions is immediate. It is obvious from the definition that for every modal circuit, there is a formula which is equivalent to the circuit up to losing syntactically irrelevant variables, i.e., variables that only appear in input-gates for which there is no path to the output-gate do not appear in the formula generated from a circuit. However, it is clear that for the questions of satisfiability and tautology that we study in this paper, these variables are not of interest.

Since every circuit therefore has an equivalent formula, it is clear that considering circuits instead of formulas does not increase the expressive power, but circuits are a succinct representation of formulas (there are families of circuits representing formulas where the size of the formula expansion is exponential in the size of the circuit).

2.3. Problem definitions

We now define the various modal satisfiability problems we are interested in. As usual in computational complexity, we define the problems as the sets of their yes-instances.

Definition 2.3. Let B be a finite set of Boolean functions, $\mathcal{F}$ a class of frames, $k \geqslant 0$, and $M \subseteq \{\Diamond, \Box\}$. Then

- $\mathrm{MFORM}^k_M(B)$ is the set of formula expansions of circuits in $\mathrm{MCIRC}^k_M(B)$, i.e., the set of modal formulas using operators from B, and modalities $\Box_1, \ldots, \Box_k$ (if $\Box \in M$) and $\Diamond_1, \ldots, \Diamond_k$ (if $\Diamond \in M$),
- $\mathcal{F}\text{-}\mathsf{FSAT}^K_M(B)$ is the set of $\mathcal{F}_k$-satisfiable formulas from $\mathrm{MFORM}^k_M(B)$,
- $\mathcal{F}\text{-}\mathsf{CSAT}^k_M(B)$ is the set of $\mathcal{F}_k$-satisfiable circuits from $\mathrm{MCIRC}^k_M(B)$,
- $\mathcal{F}\text{-}\mathsf{FTAUT}^k_M(B)$ is the set of $\mathcal{F}_k$-tautologies in $\mathrm{MFORM}^k_M(B)$,
- $\mathcal{F}\text{-}\mathsf{CTAUT}^k_M(B)$ is the set of $\mathcal{F}_k$-tautologies in $\mathrm{MCIRC}^k_M(B)$.

For readability, we often leave out the set brackets and write, for example, $\mathrm{K}\text{-}\mathsf{FSAT}^1_\Box(\oplus, 1)$ instead of $\mathrm{K}\text{-}\mathsf{FSAT}^1_{\{\Box\}}(\{\oplus, 1\})$. In addition to specifying whether $\Diamond$ and $\Box$ are allowed "globally," we could also allow our model to specify for each $i \in \{1, \ldots, k\}$ whether $\Diamond_i$ and $\Box_i$ are allowed to appear in the circuits. However, our hardness results usually require only a single one of these operators to be present (and upper complexity bounds obviously transfer to the restricted setting). Therefore, our definition captures the significant variations of the problems we study.

From the above definitions, the following is immediate, which we will often use without reference. It is obvious that analogous results hold for the tautology problem as well. Due to this proposition, it is clear that it suffices to state lower complexity bounds for the problems involving formulas, and upper bounds for the problems involving circuits.

Proposition 2.4. *Let $B_1 \subseteq B_2$ be finite sets of Boolean functions, let $\mathcal{F}$ be a class of frames, let $k_1 \leqslant k_2$, and let $M_1 \subseteq M_2 \subseteq \{\Box, \Diamond\}$. Then the following hold*:

- $\mathcal{F}\text{-}\mathsf{FSAT}^{K_1}_{M_1}(B_1) \leqslant^{\log}_m \mathcal{F}\text{-}\mathsf{FSAT}^{K_2}_{M_2}(B_2)$,
- $\mathcal{F}\text{-}\mathsf{FSAT}^{K_1}_{M_1}(B_1) \leqslant^{\log}_m \mathcal{F}\text{-}\mathsf{CSAT}^{K_2}_{M_2}(B_2)$,
- $\mathcal{F}\text{-}\mathsf{CSAT}^{K_1}_{M_1}(B_1) \leqslant^{\log}_m \mathcal{F}\text{-}\mathsf{CSAT}^{K_2}_{M_2}(B_2)$.

Table 2
Important clones and their bases.

BF	All Boolean functions
S_1	$[x \wedge \bar{y}]$
M	Monotone functions
S_{11}	$M \cap S_1$
R_1	f with $f(1,\ldots,1)=1$
D	Self-dual functions
L	Linear functions
V	Multi-ary OR and constants 0, 1
V_0	Multi-ary OR and constant 0
V_2	Multi-ary OR
E	Multi-ary AND and constants 0, 1
E_0	Multi-ary AND and constant 0
E_2	Multi-ary AND
N	Negation, identity, and constants
I	Identity and constants

Initial complexity results can be found in the literature; we state them in our notation:

Theorem 2.5. *(See [21,14].)*

(1) S5-FSAT$^1_\square(\wedge,\neg)$ *is* NP-*complete.*
(2) *Let* $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}, \mathrm{K4}, \mathrm{T}, \mathrm{S4}\}$. *Then* $\mathcal{F}$-FSAT$^1_\square(\wedge,\neg)$ *is* PSPACE-*complete.*
(3) *Let* $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}, \mathrm{K4}, \mathrm{T}, \mathrm{S4}, \mathrm{S5}\}$, *and let* $k \geqslant 2$. *Then* $\mathcal{F}$-FSAT$^k_\square(\wedge,\neg)$ *is* PSPACE-*complete.*

In [17], Hemaspaandra examined the complexity of K-FSAT$^1_M(B)$ for all $M \subseteq \{\square, \Diamond\}$ and $B \subseteq \{\wedge, \vee, \neg, 0, 1\}$. In this paper, we generalize this result in several ways: We classify the complexity of modal satisfiability for all finite sets of Boolean functions (in particular, we determine the complexity of an infinite number of problems), and we consider multi-modal logic as well. Further, we also consider the case of circuits instead of formulas, and study different frame classes. Finally, we also consider the validity problem.

2.4. Clones and Post's Lattice

The notion of clones is very helpful to bring structure to this infinite set of problems. We introduce the necessary definitions, and some important properties of Boolean functions. An n-ary function f is a *projection function* if there is some i such that for all $\alpha_1,\ldots,\alpha_n \in \{0,1\}$, $f(\alpha_1,\ldots,\alpha_n)=\alpha_i$. A set B of Boolean functions is called a *clone* if it is closed under *superposition*, that is, B contains all projection functions and is closed under arbitrary composition, i.e., if $f \in B$ and $g_1,\ldots,g_n \in B$, then $f(g_1,\ldots,g_n) \in B$. It is easy to see that the set of clones forms a lattice. Post determined the complete set of clones, as well as their inclusion structure [29]. A graphical presentation of the lattice of clones, also known as Post's Lattice, can be found in Fig. 1. For a set B of Boolean functions, let $[B]$ be the smallest clone containing B. The set B is also called the *base* of the clone $[B]$.

We briefly define the clones that arise in our complexity classification (see Table 2). The smallest clone contains only projections and is named I_2. Further, $I_1 = [\{1\}]$. The largest clone $BF = [\{\wedge,\neg\}]$ is the set of all Boolean functions. The set of all monotone functions forms a clone denoted by $M = [\{\vee,\wedge,0,1\}]$. D consists of all *self-dual* functions, i.e., $f \in D$ if and only if $f(x_1,\ldots,x_n) = \neg f(\bar{x}_1,\ldots,\bar{x}_n)$. $L = [\{\oplus,1\}]$ is the set of all linear Boolean functions (where $\oplus$ is the Boolean exclusive or). The clone of all Boolean functions that can be written using only disjunction and constants is called $V = [\{\vee,1,0\}]$; further, $V_0 = [\{\vee,0\}]$ and $V_2 = [\{\vee\}]$. Similarly, the clone $E = [\{\wedge,0,1\}]$ contains the Boolean functions that can be written as conjunctions of variables and constants; $E_0 = [\{\wedge,0\}]$ and $E_2 = [\{\wedge\}]$. R_1 is built from all 1-*reproducing* functions, i.e., all functions f satisfying $f(1,\ldots,1)=1$. The clone $N = [\{\neg,1\}]$ consists of the projections, their negations, and all constant Boolean functions. Finally, $S_1 = [\{x \wedge \bar{y}\}]$ and $S_{11} = S_1 \cap M$.

If we interpret Boolean formulas as Boolean functions, then $[B]$ consists of all propositional formulas that are equivalent to a formula built with variables and operators from B. Therefore, this framework can be used to investigate problems related to Boolean formulas depending on which connectives are allowed. Several problems have been studied in this context: Lewis proved that the satisfiability problem for Boolean formulas with connectives from B is NP-complete if $S_1 \subseteq [B]$ and in P otherwise [24]. Another example is the classification of the equivalence problem given by Reith: Deciding whether two formulas with connectives from B are equivalent is in LOGSPACE if $[B] \subseteq V$ or $[B] \subseteq E$ or $[B] \subseteq L$, and coNP-complete in all other cases [30]. Dichotomy results for counting the solutions of formulas [32], finding the minimal solutions of formulas [31], and learnability of Boolean formulas and circuits [9] were achieved as well. After the presentation of our results in [1], Bauland et al. investigated the analogous problem in the context of temporal logics [3,2].

Post's Lattice has also been a helpful tool in the constraint satisfaction context. It can be used to obtain a very easy proof of Schaefer's Theorem [33] and related complexity classifications. This is surprising, because constraint satisfaction

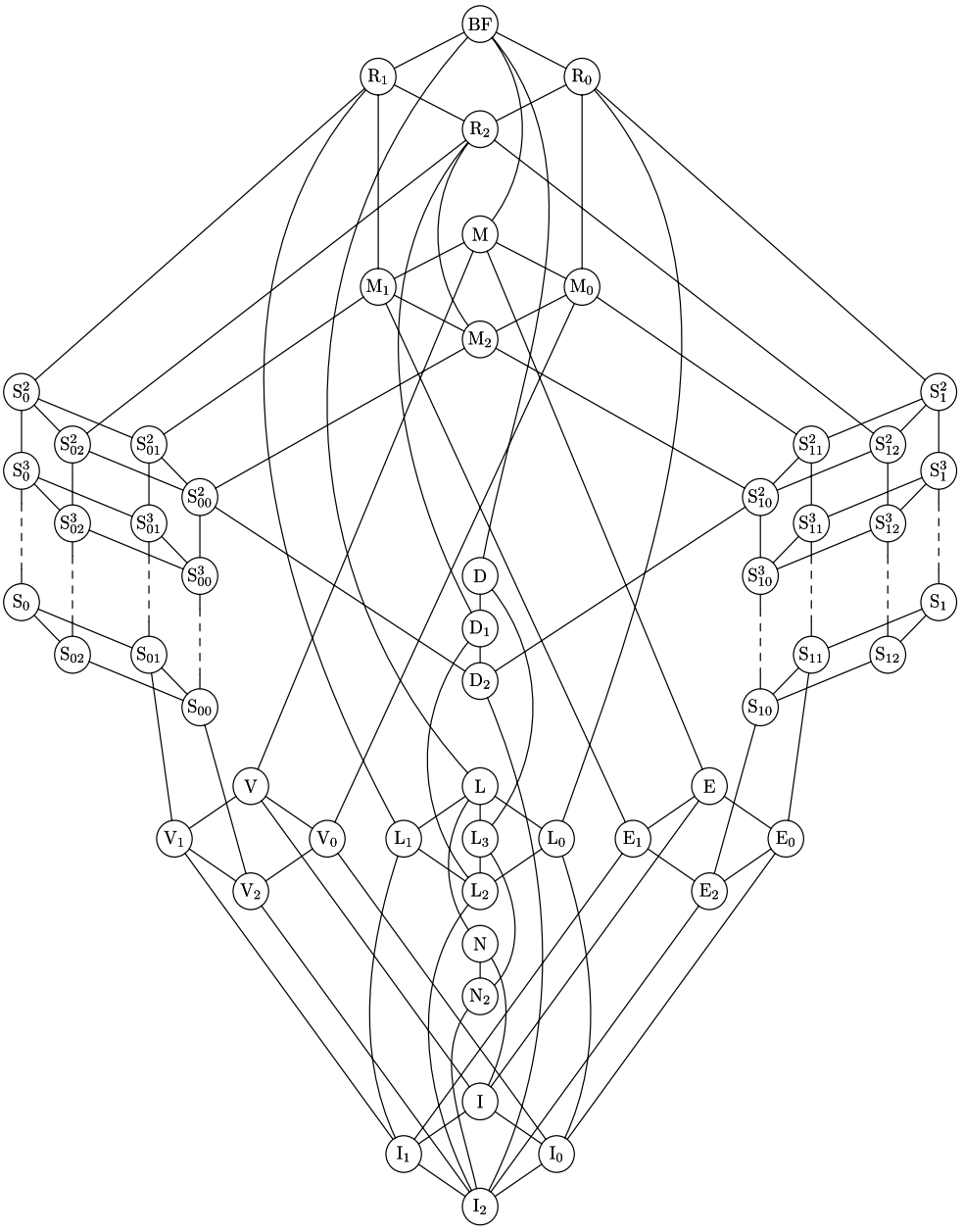

Fig. 1. Post's Lattice.

problems are not related to Post's Lattice by definition, but clones appear indirectly through a Galois connection [20]. For more information about the use of Post's Lattice in complexity classifications of propositional logic, see, for example, [6,7]. Finally, the notion of clones is not restricted to the Boolean case, but has been studied for arbitrary domains. Chapter 1 of [28] contains an introduction to the basic notions of clone theory, the monograph [23] is an excellent survey of the field.

The structure given by Post's Lattice enables us to compare the complexity of our circuit-related problems for the cases in which the corresponding clones are comparable. For circuits, we get a stronger result than Proposition 2.4: The complexity of our problems does not depend on the actual set B of Boolean functions, but just on the clone $[B]$ generated by it. Again, an analogous result holds for the tautology problem.

Lemma 2.6. *Let B_1, B_2 be finite sets of Boolean functions, $\mathcal{F}$ a class of frames, $k \geqslant 1$, and $M \subseteq \{\Diamond, \Box\}$. If $B_1 \subseteq [B_2]$, then $\mathcal{F}\text{-CSAT}_M^k(B_1) \leqslant_m^{\log} \mathcal{F}\text{-CSAT}_M^k(B_2)$.*

Proof. This reduction is achieved by replacing every occurring gate representing a function from B_1 with the appropriate B_2-circuit computing the same function. The resulting circuit obviously is $\mathcal{F}$-equivalent to the original circuit. □

It is worth noting that an analogous result for formulas cannot be obtained in such an easy way, as the following example illustrates: Consider the sets $B_1 = \{\oplus\}$ and $B_2 = \{\wedge, \vee, \neg\}$ of Boolean functions. Since every Boolean function can be represented using only AND, OR, and negation gates, it is obvious that $B_1 \subseteq [B_2]$ holds. However, a reduction from K-FSAT$^0_\emptyset(B_1)$ to K-FSAT$^0_\emptyset(B_2)$ cannot be achieved in a straightforward manner, as a formula transformation analogous to the proof of Lemma 2.6 would replace a subformula $\varphi_1 \oplus \varphi_2$ with the formula $(\varphi_1 \wedge \neg\varphi_2) \vee (\neg\varphi_1 \wedge \varphi_2)$, and repeated application of this transformation leads to exponential size for nested formulas. However, we will see that in the cases arising in this paper, the complexity of a problem $\mathcal{F}$-FSAT$^k_M(B)$ also only depends on the clone generated by B.

3. The satisfiability problem

Our main results are the classification theorems which we will present now. A graphical presentation of these results can be found in Figs. 2 and 3. For the most general problem of K-satisfiability, we get the following trichotomy:

Theorem 3.1. *Let B be a finite set of Boolean functions, $k \geqslant 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$. Then the following hold*:

- *If $B \subseteq \mathrm{R}_1, \mathrm{D}, \mathrm{V}$, or L, then* K-FSAT$^k_M(B)$, K-CSAT$^k_M(B) \in \mathrm{P}$ (*Corollary* 3.15, *Theorems* 3.18, 3.19).
- *If $\mathrm{E}_0 \subseteq [B] \subseteq \mathrm{E}$, then* K-FSAT$^k_M(B)$, K-CSAT$^k_M(B)$ *are* coNP*-complete if $M = \{\Box, \Diamond\}$, and in* P *otherwise* (*Section* 3.3, *Theorem* 3.20).
- *If $\mathrm{S}_{11} \subseteq [B] \subseteq \mathrm{M}$, then* K-FSAT$^k_M(B)$ *and* K-CSAT$^k_M(B)$ *are* PSPACE*-complete if $M = \{\Box, \Diamond\}$, and in* P *otherwise* (*Corollary* 3.11, *Theorem* 3.20).
- *Otherwise, $\mathrm{S}_1 \subseteq [B]$ and* K-FSAT$^k_M(B)$ *and* K-CSAT$^k_M(B)$ *are* PSPACE*-complete* (*Corollary* 3.11).

For the logic KD, we get the following complete classification:

Theorem 3.2. *Let B be a finite set of Boolean functions, $k \geqslant 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$. Then the following hold*:

- *If $B \subseteq \mathrm{R}_1, \mathrm{D}, \mathrm{M}$, or L, then* KD-FSAT$^k_M(B)$, KD-CSAT$^k_M(B) \in \mathrm{P}$ (*Corollary* 3.15, *Theorems* 3.16, 3.19).
- *Otherwise, $\mathrm{S}_1 \subseteq [B]$, and* KD-FSAT$^k_M(B)$ *and* KD-CSAT$^k_M(B)$ *are* PSPACE*-complete* (*Corollary* 3.11).

This dichotomy is a natural analog of Lewis's result that the satisfiability problem for Boolean formulas with connectives from B is NP-complete if $\mathrm{S}_1 \subseteq [B]$ and in P otherwise [24].

From these theorems, we conclude that using the more succinct representation of modal circuits does not increase the polynomial degree of the complexity of these satisfiability problems (for two problems A and B, we write $A \equiv^p_m B$ if $A \leqslant^p_m B$ and $B \leqslant^p_m A$).

Corollary 3.3. *Let B be a finite set of Boolean functions, $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}\}$, $k \geqslant 1$, and let $M \subseteq \{\Box, \Diamond\}$. Then $\mathcal{F}$-*CSAT$^k_M(B) \equiv^p_m$ $\mathcal{F}$-FSAT$^K_M(B)$.

The following is our classification for the logics T and S4, which gives a complete classification except for the cases where $[B]$ is one of the clones L or L_0.

Theorem 3.4. *Let B be a finite set of Boolean functions, $\mathcal{F} \in \{\mathrm{T}, \mathrm{S4}\}$, $k \geqslant 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$.*

- *If $B \subseteq \mathrm{R}_1, \mathrm{D}, \mathrm{N}$ or M, then $\mathcal{F}$-*FSAT$^K_M(B)$, $\mathcal{F}$-CSAT$^k_M(B) \in \mathrm{P}$ (*Corollary* 3.15, *Theorems* 3.16, 3.17).
- *If $\mathrm{S}_1 \subseteq [B]$, then $\mathcal{F}$-*FSAT$^K_M(B)$ *and $\mathcal{F}$-*CSAT$^k_M(B)$ *are* PSPACE*-complete* (*Theorem* 3.6, *Corollary* 3.11).
- *Otherwise, $[B] \in \{\mathrm{L}, \mathrm{L}_0\}$.*

The logic S5 behaves differently: It is well known that the satisfiability problem for this logic can be solved in NP, as long as only one modality is present [21]. As soon as at least two modalities are involved, the problem becomes PSPACE-complete [14]. We show that, in a similar way to most of the other logics with PSPACE-complete satisfiability problems that we considered, the problem is hard for this complexity class as soon as the propositional functions we allow in the formulas and circuits can express the crucial function $x \wedge \overline{y}$, which corresponds to clones that are supersets of S_1.

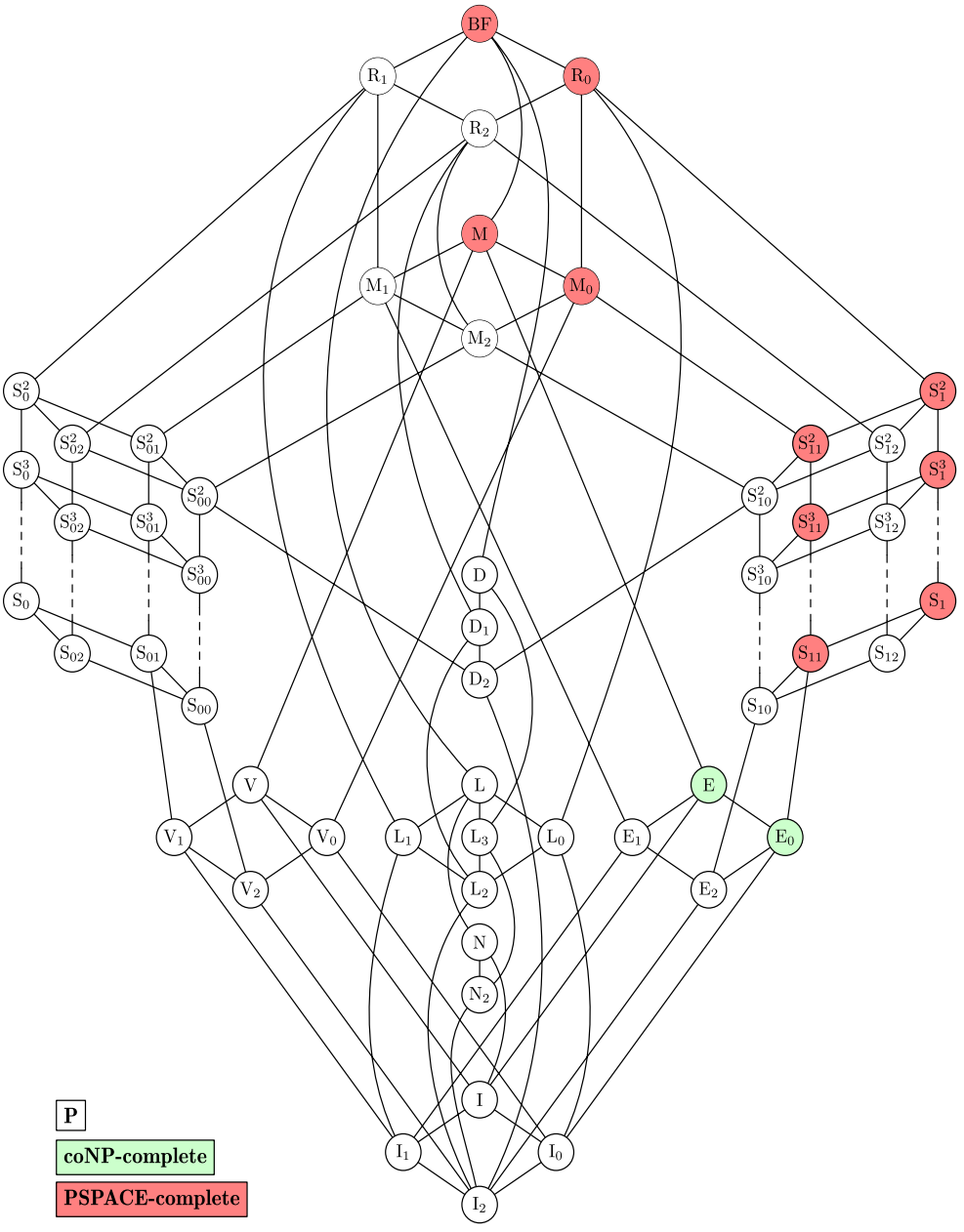

Fig. 2. The complexity of K-FSAT$^k_{\Box,\Diamond}(B)$ and K-CSAT$^k_{\Box,\Diamond}(B)$ for $k \geqslant 1$.

Theorem 3.5. *Let B be a finite set of Boolean functions, $k \geqslant 1$, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$. Then the following hold*:

- *If $B \subseteq \mathrm{R}_1$, D, N or M, then* S5-FSAT$^k_M(B)$, S5-CSAT$^k_M(B) \in$ P (*Corollary* 3.15, *Theorems* 3.16, 3.17).
- *If $\mathrm{S}_1 \subseteq [B]$, then* S5-FSAT$^k_M(B)$ *and* S5-CSAT$^k_M(B)$ *are* PSPACE*-complete if $k \geqslant 2$, and* NP*-complete if $k = 1$* (*Corollary* 3.11).
- *Otherwise, $[B] \in \{\mathrm{L}, \mathrm{L}_0\}$.*

The above classifications for logics different from K and KD leave open the cases where the set B generates one of the clones L and L_0. We will discuss these open issues in Section 3.4. Note that in the above theorem, the NP-hardness results are immediate from the previously-mentioned results in [24]: It directly follows from his result that for any non-empty class $\mathcal{F}$ of frames, the problem $\mathcal{F}$-FSAT$^0_\emptyset(B)$ is NP-hard if $\mathrm{S}_1 \subseteq [B]$.

The rest of this section is devoted to proving these theorems. As mentioned before, if suffices to prove upper bounds for circuits and lower bounds for formulas.

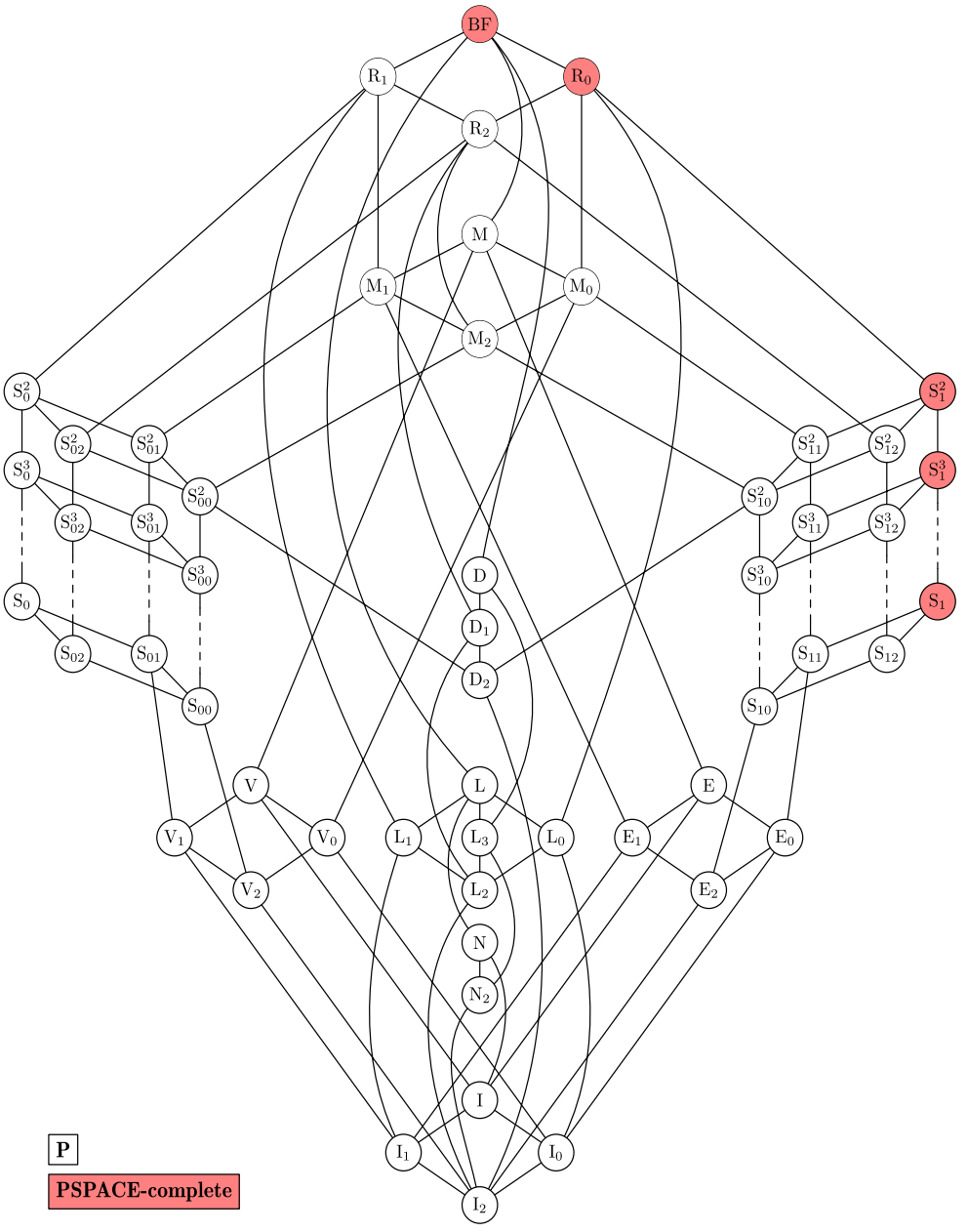

Fig. 3. The complexity of KD-FSAT$^k_M(B)$ and KD-CSAT$^k_M(B)$ for any $\emptyset \neq M \subseteq \{\Box, \Diamond\}$ and K-FSAT$^k_\Diamond(B)$, K-FSAT$^k_\Box(B)$, K-CSAT$^k_\Diamond(B)$, and K-CSAT$^k_\Box(B)$ for $k \geqslant 1$.

3.1. General upper bounds

It is well known that the $\mathcal{F}$-satisfiability problem for modal formulas using the operators $\Box, \wedge$, and $\neg$ is solvable in PSPACE for a variety of classes $\mathcal{F}$ of frames for both the uni-modal case [21] and the general multi-modal setting [14]. The proof of the following theorem shows that the circuit case can be reduced to the formula case, thus putting the circuit problems in PSPACE as well.

The intuitive reason why the complexity of our satisfiability problems does not increase significantly when considering circuits instead of formulas is that for many algorithms in modal logic, the complexity depends on the number of appearing subformulas more than on the length of the formula.

Theorem 3.6. *Let B be a finite set of Boolean functions, $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}, \mathrm{T}, \mathrm{S4}, \mathrm{S5}\}$, $k \geqslant 1$, and $M \subseteq \{\Box, \Diamond\}$. Then $\mathcal{F}$-CSAT$^k_M(B) \in$ PSPACE and* S5-CSAT$^1_M(B) \in$ NP.

Proof. The main idea of the proof is to transform the given circuit in $\mathrm{MCIRC}^k_M(B)$ into a modal formula using modal operators $\Box_1, \ldots, \Box_k$, the modal operator E (where $\mathsf{E}\varphi$ is an abbreviation for $\Box_1\varphi \wedge \cdots \wedge \Box_k\varphi$), and the propositional operators $\wedge, \vee, \neg$. Satisfiability for these formulas for the classes $\mathcal{F}$ of frames that we consider can be solved in PSPACE and the case where $\mathcal{F} = \mathrm{S5}$ and $k = 1$ can be solved in NP [21,14]. Note that their proofs do not cover the E-operator, but they work without any change if $\mathsf{E}\varphi$ is always locally evaluated as its expansion $\Box_1\varphi \wedge \cdots \wedge \Box_k\varphi$ in the algorithms presented in Section 6.3 of [14]: Their proof constructs a tableau for deciding satisfiability, where sets called $L(s)$ contain formulas which have to be true in a certain world. The only modification required for the algorithms in that paper is that when constructing the tableau, as soon as the formula $\mathsf{E}\varphi$ is a member of a set $L(s)$, then also add $\Box_1\varphi, \ldots, \Box_k\varphi$ to $L(s)$ (note that in the following, formulas of the form $\mathsf{E}\varphi$ will never appear negated, so this construction suffices).

The reduction works as follows: Let C be a circuit in $\mathrm{MCIRC}^k_M(B)$. Due to Lemma 2.6 and since PSPACE is closed under $\leqslant^{\log}_m$-reductions, we can without loss of generality assume that $B = \{\wedge, \neg\}$, and since we can express $\Diamond$ with $\Box$ and negation, we can assume that no $\Diamond$-gates appear in C. We also can assume that the set of gates of C and propositional variables appearing in C are disjoint. For every gate g in C, define $f'(C, g)$ as follows:

- If g is an input gate labeled x_i, then $f'(C, g) = g \leftrightarrow x_i$.
- If g is a $\neg$-gate, then $f'(C, g) = g \leftrightarrow \neg h$, where h is the predecessor gate of g in C.
- If g is an $\wedge$-gate, then $f'(C, g) = g \leftrightarrow (h_1 \wedge h_2)$, where h_1, h_2 are the predecessor gates of g in C.
- If g is a $\Box_i$-gate for some $1 \leqslant i \leqslant k$, then $f'(C, g) = g \leftrightarrow \Box_i h$, where h is the predecessor gate of g in C.

In this way, the gates of the circuit are represented by variables in the corresponding formula. We will view $f'(C, g)$ as a formula over $\{\Box_1, \ldots \Box_k, \wedge, \neg\}$, by viewing "$\varphi \leftrightarrow \psi$" as shorthand for "$\neg(\varphi \wedge \neg\psi) \wedge \neg(\neg\varphi \wedge \psi)$." Clearly, f' is computable in logarithmic space (note that the $\leftrightarrow$ symbols do not occur nested). We now define the actual reduction as follows: For every circuit $C \in \mathrm{MCIRC}^k_M(\wedge, \neg)$ with output gate g_{out},

$$f(C) = g_{out} \wedge \bigwedge_{g \text{ gate in } C} \bigwedge_{i=0}^{md(C)} \mathsf{E}^i f'(C, g).$$

Here $\mathsf{E}^i\varphi$ denotes $\underbrace{\mathsf{E}\ldots\mathsf{E}}_{i \text{ times}}\varphi$. Clearly, f is computable in logarithmic space. We will now show that C is $\mathcal{F}_k$-satisfiable if and only if $f(C)$ is $\mathcal{F}_k$-satisfiable.

First suppose that C is $\mathcal{F}_k$-satisfiable. Let $M = (W, R_1, \ldots, R_k, \pi)$ be an $\mathcal{F}_k$-model, and let $w_0 \in W$ be a world such that $M, w_0 \models C$. The model M' is defined over the same set of worlds with the same successor relations, and inherits the truth assignment from M for all variables appearing in C. For the new variables, the truth assignment π' of M' is defined as follows: For every gate g in C, $\pi'(g) = \{w \in W \mid M, w \models C_g\}$. Here C_g is the sub-circuit of C with output gate g (i.e., the circuit obtained from C by defining g to be the output gate, and then removing all gates from which there is no path to g). By definition of π', for every world $w \in W$ and for every gate $g \in C$, $M', w \models g$ if and only if $M', w \models C_g$. It is easy to show (see below) that for every world $w \in W$ and for every gate $g \in C$, $M', w \models f'(C, g)$. This implies that $M', w_0 \models \bigwedge_{g \text{ gate in } C} \bigwedge_{i=0}^{md(C)} \mathsf{E}^i f'(C, g)$. Since $M, w_0 \models C$ and $C = C_{g_{out}}$, it follows by the definition of π' that $M', w_0 \models g_{out}$. It follows that $M', w_0 \models f(C)$, and thus $f(C)$ is $\mathcal{F}$-satisfiable.

To be complete, we will show that, as mentioned above, for every world $w \in W$ and for every gate $g \in C$, $M', w \models f'(C, g)$. We make a case distinction.

- g is an input gate x_i. By definition of π', $M', w \models g$ if and only if $M', w \models x_i$. It follows that $M', w \models g \leftrightarrow x_i$.
- g is a $\neg$-gate. Let h be the predecessor gate of g. $M', w \models g$ if and only if $M', w \models C_g$. The latter holds if and only if $M', w \not\models C_h$. This holds if and only if $M', w \not\models h$. It follows that $M', w \models g \leftrightarrow \neg h$.
- g is an $\wedge$-gate. Let h_1 and h_2 be the predecessor gates of g. $M', w \models g$ if and only if $M', w \models C_g$. The latter holds if and only if $M', w \models C_{h_1}$ and $M', w \models C_{h_2}$. By definition of π', $M', w \models C_{h_1}$ if and only if $M', w \models h_1$ and $M', w \models C_{h_2}$ if and only if $M', w \models h_2$. It follows that $M', w \models g \leftrightarrow (h_1 \wedge h_2)$.
- g is a $\Box_i$-gate for some i. Let h be the predecessor gate of g. $M', w \models g$ if and only if $M', w \models C_g$. The latter holds if and only if $(\forall w' \in W)[wR_iw' \Rightarrow M', w' \models C_h]$. This holds if and only if $(\forall w' \in W)[wR_iw' \Rightarrow M', w' \models h]$. It follows that $M', w \models g \leftrightarrow \Box_i h$.

For the converse, suppose that $f(C)$ is $\mathcal{F}$-satisfiable. Let M be an $\mathcal{F}$-model, and let $w_0 \in W$ be a world such that $M, w_0 \models f(C)$. We will prove by induction on the structure of circuit C_g that for every gate $g \in C$ and for every world w that is reachable from w_0 in at most $md(C) - md(C_g)$ steps, $M, w \models C_g$ if and only if $M, w \models g$. This clearly implies that $M, w_0 \models C$, and thus C is $\mathcal{F}$-satisfiable.

- g is an input gate x_i. Then C_g is equivalent to x_i. Since $M, w \models g \leftrightarrow x_i$, it follows that $M, w \models C_g$ if and only if $M, w \models g$.

- g is a $\neg$-gate. Let h be the predecessor gate of g. Then $M, w \models C_g$ if and only if $M, w \not\models C_h$. By induction, the latter holds if and only if $M, w \not\models h$. Clearly, $M, w \not\models h$ if and only if $M, w \models \neg h$. Since $M, w \models g \leftrightarrow \neg h$, it follows that $M, w \models C_g$ if and only if $M, w \models g$, as required.
- g is an $\wedge$-gate. Let h_1 and h_2 be the predecessor gates of g. Then $M, w \models C_g$ if and only if $M, w \models C_{h_1}$ and $M, w \models C_{h_2}$. By induction, the latter holds if and only if $M, w \models h_1$ and $M, w \models h_2$, and this holds if and only if $M, w \models h_1 \wedge h_2$. Since $M, w \models g \leftrightarrow (h_1 \wedge h_2)$, it follows that $M, w \models C_g$ if and only if $M, w \models g$, as required.
- g is a $\Box_i$-gate for some i. Let h be the predecessor gate of g. Then $M, w \models C_g$ if and only if for all $w' \in W$ such that wR_iw', it holds that $M, w' \models C_h$. Note that $md(C_h) = md(C_g) - 1$. Since w is reachable from w_0 in at most $md(C) - md(C_g)$ steps, it follows that for every w' such that wR_iw', w' is reachable from w_0 in at most $md(C) - md(C_g) + 1 = md(C) - md(C_h)$ steps. And so, by induction, it follows that (for all $w' \in W$ such that wR_iw', it holds that $M, w' \models C_h$) if and only if (for all $w' \in W$ such that wR_iw', it holds that $M, w' \models h$), and this holds if and only if $M, w \models \Box_i h$. Since $M, w \models g \leftrightarrow \Box_i h$, it follows that $M, w \models C_g$ if and only if $M, w \models g$, as required.

Unlike K, T, S4, and S5, the logic KD is not covered in [21] or [14], therefore the above construction does not directly yield the result for KD. However, this case easily follows from the result for K, since a circuit C is KD-satisfiable if and only if $C \wedge \bigwedge_{i=0}^{md(\varphi)} \mathsf{E}^i \bigwedge_{j=1}^{k} \Diamond_j 1$ is K-satisfiable. Hence the above construction can be used to decide KD-satisfiability as well. □

Note that in the uni-modal case, we do not have to introduce the E-operator as in the proof above: By definition of the operator E, in this case $\mathsf{E}\varphi$ is equivalent to $\Box_1\varphi$. Therefore the construction of the proof directly implies that for any class $\mathcal{F}$ of frames, uni-modal satisfiability for circuits (using any set of propositional gates) is not more difficult than the satisfiability problem for $\{\wedge, \neg\}$-formulas for the same class of frames.

Corollary 3.7. *Let B be a finite set of Boolean functions and $\mathcal{F}$ a class of frames. Then $\mathcal{F}$-$\mathsf{CSAT}^1_{\Box,\Diamond}(B) \leqslant^p_m \mathcal{F}$-$\mathsf{FSAT}^1_{\Box}(\wedge, \neg)$.*

3.2. PSPACE-*completeness*

We now show how to express, in a satisfiability-preserving way, uni-modal formulas and circuits using a restricted set of Boolean connectives and one modal operator. This implies that our satisfiability problems for these restricted sets of formulas are as hard as the general case.

As mentioned in the discussion following Lemma 2.6, with many formula transformations, the size of the resulting formula can be exponential. A crucial tool in dealing with this situation is the following lemma showing that for certain sets B, there are always short formulas representing the functions AND, OR, and NOT. Part (1) is Lemma 1.4.5 from [35], the result for the case $[B] = \mathrm{BF}$ is proven in, and part (2) directly follows from the proofs in [24].

Lemma 3.8. *Let B be a finite set of Boolean functions.*

(1) *If $V \subseteq [B]$ ($E \subseteq [B]$, resp.), then there exists a B-formula $f(x, y)$ such that f represents $x \vee y$ ($x \wedge y$, resp.) and each of the variables x and y occurs exactly once in $f(x, y)$.*
(2) *If $\mathrm{N} \subseteq [B]$, then there exists a B-formula $f(x)$ such that f represents $\bar{x}$ and the variable x occurs in f only once.*

The proof of the following theorem uses a generalization of ideas from the proof for the main result in [24]. This can be applied to an arbitrary class of frames, and in particular, it yields PSPACE-completeness results for K and KD.

Theorem 3.9. *Let B be a finite set of Boolean functions such that $\mathrm{S}_1 \subseteq [B]$, $\mathcal{F}$ a class of frames, and $\emptyset \neq M \subseteq \{\Box, \Diamond\}$. Then the following hold*:

- $\mathcal{F}$-$\mathsf{FSAT}^1_{\Box,\Diamond}(\wedge, \neg) \leqslant^{\log}_m \mathcal{F}$-$\mathsf{FSAT}^1_M(B)$,
- $\mathrm{S5}$-$\mathsf{FSAT}^2_{\Box,\Diamond}(\wedge, \neg) \leqslant^{\log}_m \mathrm{S5}$-$\mathsf{FSAT}^2_M(B)$.

Proof. First consider the uni-modal case. Let $\varphi \in \mathrm{MFORM}^1_{\Box,\Diamond}(\wedge, \neg)$. Without loss of generality, assume that φ contains only modal operators from M (use the identity $\Box \equiv \neg\Diamond\neg$ otherwise). Let $B' := B \cup \{1\}$. Then Fig. 1 shows that $[B'] = \mathrm{BF}$ (since I_1 is the smallest clone containing 1, and BF is the smallest clone containing I_1 and S_1). It follows from Lemma 3.8 that there is a B'-formula $f_\neg(x)$ that represents $\bar{x}$, and x occurs in $f_\neg(x)$ only once, and there exists a B'-formula $f_\wedge(x, y)$ that represents $\wedge$ and x and y occur exactly once in $f_\wedge(x, y)$. In φ, replace every occurrence of $\wedge$ with $f_\wedge$, and every occurrence of $\neg$ with $f_\neg$. Call the resulting formula φ'. Clearly, φ' is a formula in $\mathrm{MFORM}^1_M(B')$, and φ' is equivalent to φ. By choice of $f_\vee$, $f_\wedge$, and $f_\neg$, φ' is computable in polynomial time.

Now replace every occurrence of the constant 1 with a new variable t and force t to be 1 in every relevant world by adding $\wedge \bigwedge_{i=0}^{md(\varphi)} \Box_1^i t$. This is a conjunction of linearly many terms (since $md(\varphi) \leqslant |\varphi|$). We insert parentheses in such a way that we get a tree of $\wedge$'s of logarithmic depth. Now express the $\wedge$'s in this tree with the equivalent B-formula (which exists,

since $[B] \supseteq S_1 \supset E_2 = [\wedge]$) with the result only increasing polynomially in size. It is obvious that this formula is satisfiable if and only if the original formula φ is.

Now for the bimodal case and the logic S5, we use the same construction as above, except that to force the variable t to true in all relevant worlds, we use the formula $(\Box_1\Box_2)^{md(\varphi)}t$. Due to the reflexivity of both successor relations in $S5_2$-models, this forces t to be true in all relevant worlds. □

The following theorem implies that for the logic K, PSPACE-completeness already holds for a lower class in Post's Lattice. The proof is nearly identical to the one for the above Theorem 3.9: Note that $[S_{11} \cup \{1\}] = M$, and apply Lemma 3.8 for the class M. Then follow the construction above. (We can represent $\wedge$ by a B-formula since $S_{11} \supseteq E_2 = [\wedge]$, and we can represent 0 by a B-formula since $0 \in S_{11}$.)

Theorem 3.10. *Let B be a finite set of Boolean functions such that $S_{11} \subseteq [B]$, $\mathcal{F}$ a class of frames, $k \geqslant 1$, and $M \subseteq \{\Box, \Diamond\}$. Then $\mathcal{F}\text{-FSAT}^1_M(\wedge, \vee, 0) \leqslant^{\log}_m \mathcal{F}\text{-FSAT}^1_M(B)$.*

The above theorems give the following corollary.

Corollary 3.11. *Let B be a finite set of Boolean functions, and let $\emptyset \neq M \subseteq \{\Box, \Diamond\}$.*

(1) *If $S_1 \subseteq [B]$, and $\mathcal{F}$ is a class of frames such that $S4 \subseteq \mathcal{F} \subseteq K$, and $k \geqslant 1$, then $\mathcal{F}\text{-FSAT}^k_M(B)$ and $\mathcal{F}\text{-CSAT}^k_M(B)$ are* PSPACE*-hard.*
(2) *If $S_{11} \subseteq [B]$ and $k \geqslant 1$, then $\text{K-FSAT}^k_{\Box,\Diamond}(B)$ and $\text{K-CSAT}^k_{\Box,\Diamond}(B)$ are* PSPACE*-complete.*
(3) *If $S_1 \subseteq [B]$ and $k \geqslant 2$, then $\text{S5-FSAT}^k_M(B)$ and $\text{S5-CSAT}^k_M(B)$ are* PSPACE*-complete.*

Proof. The upper bounds follow from Theorem 3.6.

(1) In [21], it is shown that for every class of frames $\mathcal{F}$ such that $S4 \subseteq \mathcal{F} \subseteq K$, the problem $\mathcal{F}\text{-FSAT}^1_M(\wedge, \neg)$ is PSPACE-hard. Therefore this follows from [21] and Theorem 3.9.
(2) In [17, Theorem 6.5], it is shown that $\text{K-FSAT}^1_{\Box,\Diamond}(\wedge, \vee, 0)$ is PSPACE-hard. Thus the result follows from Theorem 3.10.
(3) In [14], it is shown that $\text{S5-FSAT}^2_{\Box,\Diamond}(\wedge, \neg)$ is PSPACE-hard. Therefore, the result follows from Theorem 3.9. □

3.3. coNP-*completeness*

In [17], the analogous result of the following lemma was shown for uni-modal formulas. We prove that this coNP upper bound also holds for circuits.

Lemma 3.12. *Let $k \geqslant 1$. Then $\text{K-CSAT}^k_{\Box,\Diamond}(\wedge, 0, 1) \in$ coNP.*

Proof. The proof for the analogous statement for uni-modal formulas is based on the following fact: Let φ be a formula of the form $\varphi = \bigwedge_{i\in I} \Box\varphi_i^{\Box} \wedge \bigwedge_{j\in J} \Diamond\varphi_j^{\Diamond} \wedge \psi$, where I and J are finite sets of indices, $\varphi_i^{\Box}$ and $\varphi_j^{\Diamond}$ are modal formulas for all $i \in I$, $j \in J$, and ψ is a propositional formula. Then φ is satisfiable if and only if ψ is satisfiable and for every $j \in J$, $\bigwedge_{i\in I} \varphi_i^{\Box} \wedge \varphi_j^{\Diamond}$ is satisfiable [21].

This generalizes to multi-modal formulas from $\text{MFORM}^k_{\Box,\Diamond}(\wedge, 0, 1)$ in the following way: Let

$$\varphi = \bigwedge_{i\in I_1} \Box_1\varphi_i^{\Box_1} \wedge \cdots \wedge \bigwedge_{i\in I_k} \Box_k\varphi_i^{\Box_k} \wedge \bigwedge_{j\in J_1} \Diamond_1\varphi_j^{\Diamond_1} \wedge \cdots \wedge \bigwedge_{j\in J_k} \Diamond_k\varphi_j^{\Diamond_k} \wedge \psi,$$

for finite sets of indices $I_1, \ldots, I_k, J_1, \ldots, J_k$, formulas $\varphi_i^{\Box_l}, \varphi_j^{\Diamond_l} \in \text{MFORM}^k_{\Box,\Diamond}(\wedge, 0, 1)$, and a propositional $\{\wedge, 0, 1\}$-formula ψ. Then φ is satisfiable if and only if for every $1 \leqslant l \leqslant k$ and every $j \in J_l$ it holds that ψ and $\bigwedge_{i\in I_l} \varphi_i^{\Box_l} \wedge \varphi_j^{\Diamond_l}$ are satisfiable. Since every formula from $\text{MFORM}^k_{\Box,\Diamond}(\wedge, 0, 1)$ can be written in the above form and since satisfiability for the propositional part ψ can be tested in polynomial time according to [24], this leads to a recursive NP-algorithm for the question if φ is unsatisfiable.

We give an analogous proof for multi-modal circuits. Let C be a circuit from $\text{MCIRC}^k_{\Box,\Diamond}(\wedge, 0, 1)$ with output-gate *out*. If *out* is a $\Box_i$-gate for some $1 \leqslant i \leqslant k$, then φ is satisfied in every world without a successor, if *out* is a $\Diamond_i$-gate for some $1 \leqslant i \leqslant k$, then C is satisfiable if and only if the circuit obtained from C by using the predecessor of *out* as output-gate is satisfiable, and finally if *out* is an input-gate or a constant gate, then satisfiability can be tested trivially. Therefore we assume without loss of generality *out* to be an $\wedge$-gate. For a set of gates G we define $\text{pred}(G)$ to be the set of all predecessor gates of gates in G and $\wedge$-$\text{pred}(G)$ to be the set of all non-$\wedge$-gates g which are connected to G by a path from g to a gate $g' \in G$ where all gates on the path excluding g (but including g' if $g \neq g'$) are $\wedge$-gates.

For $1 \leqslant i \leqslant k$ let $G_{\Box_i}$ be the set of all $\Box_i$-gates in C, $G_{\Diamond_i}$ the set of all $\Diamond_i$-gates in C and G the set of all propositional gates in C. Then, due to the equivalence above, C is satisfiable if and only if

$$\bigwedge_{g \in \wedge\text{-pred}(\{out\}) \cap G} \varphi_g \quad \text{and} \quad \bigwedge_{g \in \text{pred}(\wedge\text{-pred}(\{out\}) \cap G_{\Box_i})} \varphi_g \wedge \bigwedge_{g \in \text{pred}(\{g_{\Diamond_i}\})} \varphi_g$$

are satisfiable for every $1 \leqslant i \leqslant k$ and every $g_{\Diamond_i} \in \wedge\text{-pred}(\{out\}) \cap G_{\Diamond_i}$, where for a gate g, the formula φ_g is defined as in the definition for modal circuits, i.e., φ_g is the formula represented by the sub-circuit with output-gate g. Note that due to the definition of $\wedge$-pred, the first of these formulas is a propositional formula.

More generally, a formula of the form $\varphi = \bigwedge_{g \in H} \varphi_g$ for a set H of gates from C is satisfiable if and only if

$$\psi := \bigwedge_{g \in \wedge\text{-pred}(H) \cap G} \varphi_g \quad \text{and} \quad \varphi^{g_{\Diamond_i}} := \bigwedge_{g \in \text{pred}(\wedge\text{-pred}(H) \cap G_{\Box_i})} \varphi_g \wedge \bigwedge_{g \in \text{pred}(\{g_{\Diamond_i}\})} \varphi_g$$

are satisfiable for every $1 \leqslant i \leqslant k$ and every $g_{\Diamond_i} \in \wedge\text{-pred}(H) \cap G_{\Diamond_i}$.

Note that ψ is a conjunction of constants and variables, therefore satisfiability of ψ can be tested in polynomial time. It is obvious that constructing the sets $\text{pred}(H)$ and $\wedge\text{-pred}(H)$ needs only polynomial time as well.

For testing if a formula φ represented by H is unsatisfiable it suffices to check if ψ is unsatisfiable, and, if this is not the case, to guess a $g_{\Diamond_i} \in \wedge\text{-pred}(H) \cap G_{\Diamond_i}$ for some $1 \leqslant i \leqslant k$ and to recursively test unsatisfiability of $\varphi^{g_{\Diamond_i}}$, which is represented by the set $\text{pred}(\wedge\text{-pred}(H) \cap G_{\Box_i}) \cup \text{pred}(\{g_{\Diamond_i}\})$. Since in every recursion the length of the longest path between an input-gate and a gate in H decreases, the algorithm stops after at most $|C|$ recursions.

Hence, starting with $H = \{out\}$ we get an NP-algorithm for testing unsatisfiability of C. □

In [18], it is shown that $\text{K-FSAT}^1_{\Box,\Diamond}(\wedge, 0)$ is coNP-hard. Applying Lemma 3.8, we obtain the following result.

Lemma 3.13. *Let B be a finite set of Boolean functions such that $\mathrm{E}_0 \subseteq [B] \subseteq \mathrm{E}$, and $k \geqslant 1$. Then $\text{K-FSAT}^k_{\Box,\Diamond}(B)$ is* coNP*-hard.*

Proof. It obviously suffices to consider the case $k = 1$. We use a similar construction as in the proof for Theorem 3.9. Let $B' := B \cup \{1\}$. From the structure of Post's Lattice, it follows that $[B'] = \mathrm{E}$. Hence, by Lemma 3.8, we have a short B'-formula for AND, and can convert $\text{MFORM}^1_{\Box,\Diamond}(\wedge, 0)$-formulas into equivalent formulas from $\text{MFORM}^1_{\Box,\Diamond}(B')$. We remove the occurrences of 1 as in Theorem 3.9: Introduce a variable t and force it to be 1 with the logarithmic tree construction. The coNP-hardness then follows from the above-mentioned result from [18]. □

3.4. Polynomial time

We now give our polynomial-time algorithms. We will see that in many of those cases where the restriction of the propositional operators to a certain set B leads to a polynomial-time decision procedure in the propositional case, the same is true for the corresponding modal problems. One notable exception is the case of monotone formulas: For propositional monotone formulas, satisfiability can easily be tested, since such a formula is satisfiable if and only if it is satisfied by the constant 1-assignment. For modal satisfiability, we have seen in Corollary 3.11 that the corresponding problem is as hard as the standard satisfiability problem for modal logic. The other exception concerns formulas using only conjunction and constants: As a special case of monotone formulas, satisfiability testing is easy for propositional logic. However, Section 3.3 showed that the problem is coNP-complete for modal logic.

Lemma 3.14. *Let B be a finite set of Boolean functions, $k \geqslant 1$, and $\varphi \in \text{MFORM}^k_{\Box,\Diamond}(B)$. If the formula φ^{id}, which is obtained by changing every modal operator in φ to the identity, is satisfiable, then φ is satisfiable in the reflexive singleton.*

Proof. Let I be a propositional assignment satisfying φ^{id}. Let M be the model consisting of the reflexive singleton, where each variable is true if and only if it is true in I. Since in this model, every modal operator can only refer to the same single world in the model, the operators are equivalent to the identity function, implying the result. □

It is obvious that every propositional B-formula for $B \subseteq \mathrm{R}_1$ or $B \subseteq \mathrm{D}$ is satisfiable [24]: In the first case, the all-1-assignment always is a model. In the second case, exactly one of the two constant assignments is. Hence, Lemma 3.14 immediately gives the following complexity result:

Corollary 3.15. *Let B be a finite set of Boolean functions such that $B \subseteq \mathrm{R}_1$ or $B \subseteq \mathrm{D}$, $\mathcal{F}$ a class of frames containing the reflexive singleton, and $k \geqslant 1$. Then every formula from $\text{MFORM}^k_{\Box,\Diamond}(B)$ is $\mathcal{F}$-satisfiable. In particular, $\mathcal{F}\text{-CSAT}^k_{\Box,\Diamond}(B) \in \mathrm{P}$ for $\mathcal{F} \in$ {K, KD, K4, T, S4, S5}.*

While K-satisfiability for variable-free formulas using constants, the Boolean connectives $\wedge$ and $\vee$, and both modal operators is complete for PSPACE [17], this problem (even with variables) is solvable in polynomial time if we look only at frames in which each world has a successor.

Theorem 3.16. *Let B be a finite set of Boolean functions such that $B \subseteq \mathrm{M}$, $\mathcal{F}$ a class of frames such that $\mathcal{F} \subseteq \mathrm{KD}$, and $k \geqslant 1$. Then $\mathcal{F}$-$\mathsf{CSAT}^k_{\Box,\Diamond}(B) \in \mathrm{P}$. In particular,* KD-$\mathsf{CSAT}^k_{\Box,\Diamond}(B)$, T-$\mathsf{CSAT}^k_{\Box,\Diamond}(B)$, S4-$\mathsf{CSAT}^k_{\Box,\Diamond}(B)$, S5-$\mathsf{CSAT}^k_{\Box,\Diamond}(B) \in \mathrm{P}$.

Proof. The claim is obvious if $\mathcal{F}$ is empty, hence assume that this is not the case. Let M be an $\mathcal{F}$-model, let w be a world from M, and let M_1 be the multi-modal reflexive singleton with k successor relations in which every variable is set to 1. It is easy to show by induction on the construction of any $C \in \mathrm{MCIRC}^k_M(B)$ that if $M, w \models C$, then $M_1, w \models C$ holds as well. On the other hand, if $M_1, w \models C$, then $M', w \models C$, where M' is obtained from the model M by setting every variable to true in every world. Hence, C is $\mathcal{F}$-satisfiable if and only if C is satisfied in M_1. The latter condition can obviously be verified in polynomial time. □

In the case where all of our propositional operators are essentially unary (i.e., they depend on exactly one of their arguments) or constant, we can use simple transformations to decide satisfiability, as the following theorem shows.

Theorem 3.17. *Let B be a finite set of Boolean functions such that $B \subseteq \mathrm{N}$, $\mathcal{F}$ a class of frames such that $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}, \mathrm{S4}, \mathrm{S5}, \mathrm{K4}, \mathrm{T}\}$, and $k \geqslant 1$. Then $\mathcal{F}$-$\mathsf{CSAT}^k_{\Box,\Diamond}(B) \in \mathrm{P}$.*

Proof. Since the clone N is generated by negation and the constants, we can, due to Lemma 2.6, assume that B only contains these functions. Now, let C be a circuit from $\mathrm{MCIRC}^k_M(B)$. Since every function in B is unary or constant, we can regard C as a linear graph, and thus as a formula. Using the equivalence $\Diamond_i \equiv \neg\Box_i\neg$, we can move negations inward, until we have a formula of the form $O_1 \dots O_n z$, where the O_i are modal operators, and z is either a literal or a constant. It is obvious that this formula is satisfiable if and only if z is not the constant 0, or if $\mathcal{F} = \mathrm{K}$, and there is at least one $\Box$-operator present. The transformation obviously can be performed in polynomial time. □

For monotone functions and most classes of frames that we are interested in, we already showed that the satisfiability problem can be solved in polynomial time. For the most general class of frames K, this problem is PSPACE-complete (Corollary 3.11), but a further restriction of the propositional base gives polynomial-time results here as well.

Theorem 3.18. *Let B be a finite set of Boolean functions such that $B \subseteq \mathrm{V}$, $\mathcal{F}$ a class of frames such that $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}, \mathrm{S4}, \mathrm{S5}, \mathrm{K4}, \mathrm{T}\}$, and $k \geqslant 1$. Then $\mathcal{F}$-$\mathsf{CSAT}^k_{\Box,\Diamond}(B) \in \mathrm{P}$.*

Proof. Since the clone V is generated by binary OR and the constants, we can, due to Lemma 2.6, assume that B only contains these functions. We first consider the case $\mathcal{F} \in \{\mathrm{K}, \mathrm{K4}\}$.

Let C be a circuit from $\mathrm{MCIRC}^k_{\Box,\Diamond}(B)$. If the output gate g of C is an $\vee$-gate, with predecessors h_1 and h_2 in C, then C is $\mathcal{F}$-satisfiable if and only if at least one of C_{h_1} and C_{h_2} is. If g is a $\Diamond_i$-gate with predecessor h, then C is $\mathcal{F}$-satisfiable if and only if C_h is. Finally, if g is a $\Box_i$-gate, then C is K-satisfiable.

This gives a recursive polynomial-time procedure to decide the satisfiability problem. For the classes other than K and K4, we can use the same procedure, with one modification: here, if g is a $\Box_i$-gate, then C is satisfiable if and only if C_h is satisfiable, where h is the predecessor of g in C. □

We now show that for the logics K and KD, the modal satisfiability problems for formulas having only $\oplus$ and constants in the propositional base are easy. For the propositional case, this holds because unsatisfiable formulas using only these connectives are of a very easy form: Every variable and the constant 1 appear an even number of times (see, e.g., [24]). We will see that in the modal case, unsatisfiable circuits can in polynomial time be simplified to an equivalent propositional formula.

Theorem 3.19. *Let B be a finite set of Boolean functions such that $B \subseteq \mathrm{L}$, $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}\}$ a class of frames, and $k \geqslant 1$. Then $\mathcal{F}$-$\mathsf{CSAT}^k_{\Box,\Diamond}(B) \in \mathrm{P}$.*

Proof. To prove this theorem, we present a polynomial-time algorithm $\mathrm{EQ}_{\mathcal{F}}$ to determine whether two circuits C_1 and C_2 are equivalent. This proves the theorem, since C is satisfiable if and only if C is not equivalent to 0. Because of Lemma 2.6, we can restrict ourselves to circuits from $\mathrm{MCIRC}^k_{\Box,\Diamond}(\oplus, 0, 1)$. Also note that using $\Diamond_i, \oplus$, and the constant 1, we can express $\Box_i$, and therefore it is sufficient to consider circuits in which only $\Diamond_i$-operators occur, i.e., we only need to deal with circuits from $\mathrm{MCIRC}^k_{\Diamond}(\oplus, 0, 1)$. For $\mathcal{C}$ a set of circuits, we write $\bigoplus \mathcal{C}$ to denote $\bigoplus_{C \in \mathcal{C}} C$, and we define $\bigoplus \emptyset = 0$.

Let C_1 and C_2 be circuits in $\mathrm{MCIRC}^k_{\Diamond}(\oplus, 0, 1)$. We will show that $\mathrm{EQ}_{\mathcal{F}}(C_1, C_2)$ accepts if and only if $C_1 \equiv_{\mathcal{F}} C_2$, where $\mathrm{EQ}_{\mathcal{F}}(C_1, C_2)$ is defined as follows.

(1) Write $C_1 \oplus C_2$ as $\bigoplus \mathcal{C}$ where $\mathcal{C}$ is a set of sub-circuits of $C_1 \oplus C_2$ such that for every $C \in \mathcal{C}$, C's output gate is not $\oplus$.
(2) For every $\Diamond_i D \in \mathcal{C}$ such that $\mathrm{EQ}_{\mathcal{F}}(D, 0)$, remove $\Diamond_i D$ from $\mathcal{C}$.
(3) If $\mathcal{F} = \mathrm{KD}$, for every $\Diamond_i D \in \mathcal{C}$ such that $\mathrm{EQ}_{\mathcal{F}}(D, 1)$, replace $\Diamond_i D$ by 1 in $\mathcal{C}$.
(4) For every distinct pair $(\Diamond_i D_1, \Diamond_i D_2)$ in $\mathcal{C}$ such that $\mathrm{EQ}_{\mathcal{F}}(D_1, D_2)$, remove $\Diamond_i D_1$ and $\Diamond_i D_2$ from $\mathcal{C}$.
(5) Accept if and only if $\mathcal{C}$ is propositional and $\bigoplus \mathcal{C}$ is not satisfiable.

To show that $\mathrm{EQ}_{\mathcal{F}}$ can be implemented in polynomial time, we first argue that each step in the algorithm (not counting the time for the recursive calls) can be performed in polynomial time. This is immediate for steps (2)–(4). The last step is in polynomial time because satisfiability for propositional circuits over $\{\oplus, 0, 1\}$ can be determined in polynomial time [24]. The only non-obvious step is the first. This can be done as follows. Let $\mathcal{C}'$ be the set of all sub-circuits C of $C_1 \oplus C_2$ such that the output gate of C is not $\oplus$ and such that there exists a path from the output gate of C to the output gate of $C_1 \oplus C_2$ and every other gate on this path is $\oplus$. Determine which of the elements of $\mathcal{C}'$ are relevant for the function calculated by the circuit (this can be done in polynomial time by simulation, or by using the fact that these are exactly the elements of $\mathcal{C}'$ that are connected to the output gate of $C_1 \oplus C_2$ by an odd number of $\oplus$-paths). $\mathcal{C}$ consists of all these elements. Note that not all of the elements of $\mathcal{C}'$ necessarily appear in $\mathcal{C}$.

Since all recursive calls in $\mathrm{EQ}_{\mathcal{F}}(C_1, C_2)$ have arguments that are sub-circuits of $C_1 \oplus C_2 \cup \{0, 1\}$, the algorithm can be implemented to run in polynomial time, using memoization.

It remains to show that $\mathrm{EQ}_{\mathcal{F}}(C_1, C_2)$ accepts if and only if $C_1 \equiv_{\mathcal{F}} C_2$. It is immediate that during the algorithm, $C_1 \oplus C_2 \equiv_{\mathcal{F}} \bigoplus \mathcal{C}$. And so, if $\mathrm{EQ}_{\mathcal{F}}(C_1, C_2)$ accepts, $C_1 \equiv_{\mathcal{F}} C_2$.

It remains to show that for all C_1, C_2, if $C_1 \equiv_{\mathcal{F}} C_2$, then $\mathrm{EQ}_{\mathcal{F}}(C_1, C_2)$ accepts. Suppose for a contradiction that this is not the case, and let C_1 and C_2 be two $\mathcal{F}$-equivalent circuits for which $\mathrm{EQ}_{\mathcal{F}}(C_1, C_2)$ does not accept and let (C_1, C_2) be minimal with respect to the modal depth of $C_1 \oplus C_2$.

Consider the set $\mathcal{C}$ at the last step of the algorithm. We have that $C_1 \oplus C_2 \equiv_{\mathcal{F}} \bigoplus \mathcal{C}$, $\mathcal{C}$ is not propositional (since $\mathcal{C} \equiv_{\mathcal{F}} C_1 \oplus C_2$, it follows that if φ is propositional, then φ is unsatisfiable, and hence the algorithm would accept), for every $\Diamond_i D \in \mathcal{C}$, $D \not\equiv_{\mathcal{F}} 0$, and, if $\mathcal{F} = \mathrm{KD}$, $D \not\equiv_{\mathcal{F}} 1$, and if $(\Diamond_i D_1, \Diamond_i D_2)$ is a pair in $\mathcal{C}$, then $D_1 \not\equiv_{\mathcal{F}} D_2$.

For $\mathcal{F} \in \{\mathrm{K}, \mathrm{KD}\}$ we will construct two $\mathcal{F}$-models such that exactly one of the models satisfies $\bigoplus \mathcal{C}$. This contradicts our assumption that $C_1 \equiv_{\mathcal{F}} C_2$.

Let $i \in \{1, \ldots, k\}$ be such that there exists a circuit $\Diamond_i D \in \mathcal{C}$. Let $\mathcal{D}_i = \{D \mid \Diamond_i D \in \mathcal{C}\}$ and let D_m be a minimal element of $\mathcal{D}_i$ with respect to $\mathcal{F}$-implication. This exists because $\mathcal{F}$-implication defines a partial order on the elements of $\mathcal{D}_i$ (the $\mathcal{F}$-inequivalence of elements of $\mathcal{D}_i$ ensures the anti-symmetry). For every D such that $\Diamond_j D \in \mathcal{C}$, let M_D be an $\mathcal{F}$-model and w_D be a world in M_D such that $M_D, w_D \models D$. If $\Diamond_i D \in \mathcal{C}$ and $D \neq D_m$, then make sure that $M_D, w_D \models D \wedge \neg D_m$.

Let M_1 be the model that consists of the disjoint union of all the M_D models plus an extra world w. Set all variables in w to true, and for all $j \in \{1, \ldots, k\}$, let $w R_j w_D$ if and only if ($\Diamond_j D \in \mathcal{C}$ and $\Diamond_j D \neq \Diamond_i D_m$).

Note that $M_1, w \models \Diamond_j D$ for all $\Diamond_j D \in \mathcal{C}$ such that $\Diamond_j D \neq \Diamond_i D_m$ and that $M_1, w \models \neg\Diamond_i D_m$. Let M_2 be constructed from M_1 by adding $w R_j w_{D_m}$ for all $j \in \{1, \ldots, k\}$. Then $M_2, w \models \Diamond_j D$ for all $\Diamond_j D \in \mathcal{C}$, and so $M_1, w \models \bigoplus \mathcal{C}$ if and only if $M_2, w \not\models \bigoplus \mathcal{C}$. This proves the theorem for $\mathcal{F} = \mathrm{K}$. We now turn to the case that $\mathcal{F} = \mathrm{KD}$. Note that M_2 is a KD-model, but that M_1 is not necessarily a KD-model. However, this can easily be fixed. Let M' be a KD-model and w' be a world such that $M', w' \models \neg D_m$ (such a model exists since D_m is not a KD-tautology). Let M_3 be the disjoint union of M_1 and M' and let $w R_j w'$ for all $j \in \{1, \ldots, k\}$. Then M_3 is a KD-model and $M_3, w \models \bigoplus \mathcal{C}$ if and only if $M_1, w \models \bigoplus \mathcal{C}$. □

The above proof does not generalize to classes of frames involving, for example, reflexivity or transitivity. While we conjecture that the corresponding problem for these classes of frames can still be solved in polynomial time, we mention that there are examples in the literature that behave differently—sometimes, restricting the class of frames increases the complexity of the modal satisfiability problem. For example, Halpern showed that when considering only formulas of bounded modal nesting degree, the complexity of the satisfiability problem for K drops from PSPACE-complete to NP-complete. On the other hand, for the logic S4, the problem remains PSPACE-complete [13]. Therefore syntactical restrictions that reduce the complexity of the general logic K do not necessarily also reduce the complexity for logics defined over a restricted class of models.

Our results for linear propositional functions conclude our discussion about the modal satisfiability problem for the classes of frames K and KD in the case that we allow both modal operators in our formulas and circuits: Fig. 1 shows that we have covered all clones, and hence reached a complete classification of this problem.

3.5. Satisfiability with only one type of modal operator

We now look at satisfiability problems with only one of type of operators $\Diamond$ or $\Box$ present. For sets B such that $\mathrm{S}_1 \subseteq [B]$, we already established PSPACE-completeness for the classes of frames we consider (Corollary 3.11). Since polynomial-time results for the case where we allow both $\Box$ and $\Diamond$ obviously carry over to the case where only one of them is allowed, the following theorem completes a full classification of the problem.

Theorem 3.20. *Let B be a finite set of Boolean functions such that $B \subseteq \mathrm{M}$, let $k \geqslant 0$, let $M = \{\Box\}$ or $M = \{\Diamond\}$, and let $\mathcal{F} \in \{\mathrm{K}, \mathrm{K4}\}$. Then* $\text{K-CSAT}^k_M(B) \in \mathrm{P}$.

Proof. Due to Lemma 2.6, we can assume that $B = \{\wedge, \vee, 0, 1\}$. We now show that in the case $M = \{\Diamond\}$, a circuit $C \in \mathrm{MCIRC}^k_\Diamond(B)$ is $\mathcal{F}$-satisfiable if and only if it is satisfied in the reflexive singleton where each variable is set to true, and in the case $M = \{\Box\}$, every $C \in \mathrm{MCIRC}^k_\Box(B)$ is $\mathcal{F}$-satisfiable if and only if it is satisfied in the irreflexive singleton with every variable set to true (since both the reflexive and the irreflexive singleton are $\mathcal{F}$-models, the "if" direction of this claim is trivial). These conditions obviously can be tested in polynomial time.

We show the claim by induction on the structure of the formula expansion of the circuit. If C is a single variable or a constant, then the claim obviously holds. Now assume that $C = C_1 \vee C_2$. If C is satisfiable, then at least one of C_1, C_2 is satisfiable, and due to induction, they are satisfied in the reflexive resp. irreflexive singleton with every variable set to true. If $C = C_1 \wedge C_2$, and C is satisfiable then both C_1 and C_2 are satisfiable. By induction, both of them are satisfied in the singleton with every variable set to true. Hence, C is satisfied in this singleton as well.

For the modal operators, assume that $C = \Diamond_i D$ for some $i \in \{1, \ldots, k\}$. If C is satisfiable, then obviously D is satisfiable as well, and by induction, D is satisfiable in the reflexive singleton with every variable set to true. For this case, C obviously is satisfied in the same model.

Finally, if $C = \Box_i D$ for some $i \in \{1, \ldots, k\}$, then by definition C is satisfied in the irreflexive singleton with every variable set to true. □

4. The validity problem

Besides the satisfiability problem, another problem which often is of interest is the validity problem, i.e., the problem to decide whether a given formula is valid, or is a tautology in a given logic. Recall that in our context, a formula φ is an $\mathcal{F}$-tautology if and only if φ is $\mathcal{F}$-equivalent to 1 (this is the case if and only if φ holds in every world of every $\mathcal{F}$-model).

It is obvious that a formula φ is a tautology if and only if $\neg\varphi$ is not satisfiable. With this easy observation, the complexity of the satisfiability problem and that of the validity problem often can be related to each other—they are "duals" of each other. However, in the case of restricted propositional bases, we cannot always express negation, which is necessary in order to do the transformation mentioned above directly. Therefore, we consider a more general notion of duality, which is closely related to the self-dual property defined for functions earlier: A function f is self-dual if and only if $\mathrm{dual}(f) = f$.

Definition 4.1. Let f be an n-ary Boolean function. Then $\mathrm{dual}(f)$ is the n-ary Boolean function defined as $\mathrm{dual}(f)(x_1, \ldots, x_n) = \neg f(\bar{x}_1, \ldots, \bar{x}_n)$.

For a set B of Boolean functions, $\mathrm{dual}(B)$ is defined as the set $\{\mathrm{dual}(f) \mid f \in B\}$. Obviously, a similar duality exists between the modal operators $\Diamond$ and $\Box$: For a set $M \subseteq \{\Box, \Diamond\}$, we define $\mathrm{dual}(M)$ to be the set such that $\Box \in \mathrm{dual}(M)$ if and only if $\Diamond \in M$, and $\Diamond \in \mathrm{dual}(M)$ if and only if $\Box \in M$. For a clone B, the dual clone $\mathrm{dual}(B)$ can easily be identified in Post's Lattice (see Fig. 1), as it is simply the "mirror class" with regard to the vertical symmetry axis in the lattice. The following theorem shows that complexity classifications for the satisfiability problem immediately give dual classifications for the validity problem. Note that this theorem, together with the results in Section 3.3, imply that the tautology problem for modal circuits using conjunction, constants, and both types of modal gates is NP-complete, which is an unusual complexity result for a tautology problem.

Theorem 4.2. *Let B be a finite set of Boolean functions, let $k \geqslant 0$, let $\mathcal{F}$ be a class of frames, and let $M \subseteq \{\Box, \Diamond\}$. Then the following hold*:

(1) $\mathcal{F}\text{-CTAUT}^k_M(B) \equiv^{\log}_m \overline{\mathcal{F}\text{-CSAT}^k_{\mathrm{dual}(M)}(\mathrm{dual}(B))}$.
(2) $\mathcal{F}\text{-FTAUT}^k_M(B) \equiv^{\log}_m \overline{\mathcal{F}\text{-FSAT}^k_{\mathrm{dual}(M)}(\mathrm{dual}(B))}$.

Proof. Let C be a circuit from $\mathrm{MCIRC}^k_M(B)$. We construct the circuit $\mathrm{dual}(C)$ by exchanging every f-gate for a function $f \in B$ with a $\mathrm{dual}(f)$-gate. Similarly, we replace every $\Box_i$-gate with a $\Diamond_i$-gate, and vice versa. It is obvious that this transformation can be performed in logarithmic space, and that the same transformation can be applied to formulas.

It remains to prove that C is unsatisfiable if and only if $\mathrm{dual}(C)$ is a tautology. Since dual is obviously injective, and $\mathrm{dual}(\mathrm{dual}(C)) = C$, this also proves that C is a tautology if and only if $\mathrm{dual}(C)$ is unsatisfiable, and hence proves the reduction.

Inductively, we show a more general statement: For any modal model M, let $\neg M$ denote the model obtained from M by reversing the propositional truth assignment, i.e., where a variable in a world is true if and only if the same variable is false in the same world in M. We show that for any model M and any world $w \in M$, it holds that $M, w \models C$ if and only if $\neg M, w \not\models \mathrm{dual}(C)$. This obviously completes the proof, since M is an $\mathcal{F}$-model if and only if $\neg M$ is.

We show the claim by induction on the structure of C. First, assume that C is equivalent to the variable x_i. Then $M, w \models C$ if and only if $M, w \models x_i$ if and only if $\neg M, w \not\models x_i$. Since $\mathrm{dual}(x_i) = x_i$, this proves the base step.

Now assume that the output gate g of C is an f-gate for an n-ary Boolean function $f \in B$, and let $g_1, \ldots, g_n$ be the predecessor gates of g in C. By induction, we know that for each $j \in \{1, \ldots, n\}$, it holds that $M, w \models C_{g_j}$ if and only

if $\neg M, w \not\models \mathrm{dual}(C_{g_j})$ (where C_{g_j} is the sub-circuit of C with output gate g_j). For $j \in \{1, \ldots, n\}$, let α_j be defined as 1 if $M, w \models C_{g_j}$, and 0 otherwise. By induction, we know that α_j is 1 if and only if $\neg M, w \not\models \mathrm{dual}(C)$. Now observe that $M, w \models C$ if and only if $f(\alpha_1, \ldots, \alpha_n) = 1$, if and only if $\mathrm{dual}(f)(\overline{\alpha_1}, \ldots, \overline{\alpha_n}) = 0$, and this is the case if and only if $\neg M, w \not\models \mathrm{dual}(C)$.

Now assume that the output gate g of C is a $\diamond_i$-gate for some $i \in \{1, \ldots, k\}$, and let h be the predecessor gate of g in C. Then the following holds:

$$\begin{aligned} M, w \models C \quad &\text{iff} \quad \text{there is a world } w' \text{ such that } (w, w') \in R_i \text{ and } M, w' \models C_h, \\ &\text{iff} \quad \text{there is a world } w' \text{ such that } (w, w') \in R_i \text{ and } \neg M, w' \not\models \mathrm{dual}(C_h), \\ &\text{iff} \quad \neg M, w \not\models \Box_i \mathrm{dual}(C_h), \\ &\text{iff} \quad \neg M, w \not\models \mathrm{dual}(C). \end{aligned}$$

Finally, assume that the output gate g of C is a $\Box_i$-gate for some $i \in \{1, \ldots, k\}$, and let h be the predecessor gate of g in C. Then the following holds:

$$\begin{aligned} M, w \models C \quad &\text{iff} \quad \text{for each world } w' \text{ such that } (w, w') \in R_i,\ M, w' \models C_h, \\ &\text{iff} \quad \text{for each world } w' \text{ such that } (w, w') \in R_i,\ \neg M, w' \not\models \mathrm{dual}(C_h), \\ &\text{iff} \quad \neg M, w \not\models \diamond_i \mathrm{dual}(C_h), \\ &\text{iff} \quad \neg M, w \not\models \mathrm{dual}(C). \end{aligned}$$

This concludes the induction, and therefore the proof. □

5. Conclusion and further research

We completely classified the complexity of the modal satisfiability and validity problems arising when restricting the allowed propositional operators in the formula for the logics K and KD. We showed that the more succinct representation of modal formulas as circuits does not have an effect on the complexity of these problems up to $\leqslant_m^p$-degree. We also showed that for multi-modal logics, the results only depend on whether we have 0, 1, or 2 modalities—adding more modal operators does not increase the complexity of the problems we studied. Note that in many cases, our results hold for more general classes of frames, as often they are stated for any class containing the reflexive singleton, or similar conditions. This does not only apply to most of our polynomial-time results, but also for our circuit-to-formula construction in Corollary 3.7, and our implementation results in Theorem 3.10 and the uni-modal version of Theorem 3.9.

The most obvious next question to look at is to complete our complexity classification for other classes of frames. For $\mathcal{F} \in \{\mathrm{T}, \mathrm{S4}, \mathrm{S5}\}$, our proofs already give a complete classification with the exception of the complexity of the problems $\mathcal{F}\text{-}\mathsf{FSAT}_M^K(B)$ and $\mathcal{F}\text{-}\mathsf{CSAT}_M^k(B)$ where $[B] \in \{\mathrm{L}_0, \mathrm{L}_1\}$. We conjecture that these cases are solvable in polynomial time as well, however, to solve these cases different ideas from the ones used in the proof for K and KD are required. Another interesting question is the exact complexity of our polynomial cases, most notably the case where the propositional operators represent linear functions.

There are many other interesting directions for future research. For example, one can look at other decision problems (e.g., global satisfiability and minimization), and one can try to generalize modal logic modally as well as propositionally.

Acknowledgments

We thank Michael Bauland for his work on [1], and Thomas Schneider and Heribert Vollmer for helpful discussions. We also thank the anonymous referees for their helpful comments and suggestions, and Steffen Reith for providing the figure of Post's Lattice.

References

[1] M. Bauland, E. Hemaspaandra, H. Schnoor, I. Schnoor, Generalized modal satisfiability, in: Proceedings of the 23rd Symposium on Theoretical Aspects of Computer Science, Springer, 2006, pp. 500–511.
[2] M. Bauland, M. Mundhenk, T. Schneider, H. Schnoor, I. Schnoor, H. Vollmer, The tractability of model-checking for LTL: The good, the bad, and the ugly fragments, Electr. Notes Theor. Comput. Sci. 231 (2009) 277–292.
[3] M. Bauland, T. Schneider, H. Schnoor, I. Schnoor, H. Vollmer, The complexity of generalized satisfiability for linear temporal logic, in: Foundations of Software Science and Computational Structures, Springer, 2007, pp. 48–62.
[4] B. Bennett, A. Galton, A unifying semantics for time and events, Artificial Intelligence 153 (1–2) (2004) 13–48.
[5] P. Blackburn, M. de Rijke, Y. Venema, Modal Logic, Cambridge University Press, New York, 2001.
[6] E. Böhler, N. Creignou, S. Reith, H. Vollmer, Playing with Boolean blocks, part I: Post's lattice with applications to complexity theory, SIGACT News 34 (4) (2003) 38–52.
[7] E. Böhler, N. Creignou, S. Reith, H. Vollmer, Playing with Boolean blocks, part II: Constraint satisfaction problems, SIGACT News 35 (1) (2004) 22–35.

[8] T. Coffey, R. Dojen, T. Flanagan, On the automated implementation of modal logics used to verify security protocols, in: ISICT '03: Proceedings of the 1st International Symposium on Information and Communication Technologies, Trinity College Dublin, 2003, pp. 329–334.
[9] V. Dalmau, Computational complexity of problems over generalized formulas, PhD thesis, Universitat Politécnica de Catalunya, 2000.
[10] F. Donini, B. Hollunder, M. Lenzerini, D. Nardi, W. Nutt, A. Spaccamela, The complexity of existential quantification in concept languages, Artificial Intelligence 53 (2–3) (1992) 309–327.
[11] F. Donini, M. Lenzerini, D. Nardi, W. Nutt, The complexity of concept languages, Inform. and Comput. 134 (1997) 1–58.
[12] U. Frendrup, H. Hüttel, J.N. Jensen, Modal logics for cryptographic processes, Electron. Notes Theor. Comput. Sci. 68 (2) (2002) 124–141.
[13] J. Halpern, The effect of bounding the number of primitive propositions and the depth of nesting on the complexity of modal logic, Artificial Intelligence 75 (2) (1995) 361–372.
[14] J. Halpern, Y. Moses, A guide to completeness and complexity for modal logics of knowledge and belief, Artificial Intelligence 54 (2) (1992) 319–379.
[15] J. Halpern, Y. Moses, M. Tuttle, A knowledge-based analysis of zero knowledge, in: Proceedings of the 20th Annual ACM Symposium on Theory of Computing, ACM, 1988, pp. 132–147.
[16] E. Hemaspaandra, The price of universality, Notre Dame J. Formal Logic 37 (2) (1996) 174–203.
[17] E. Hemaspaandra, The complexity of poor man's logic, J. Logic Comput. 11 (4) (2001) 609–622, corrected version: Hemaspaandra [18].
[18] E. Hemaspaandra, The complexity of poor man's logic, Tech. Rep., cs.LO/9911014v2, Computing Research Repository (CoRR), 2005.
[19] E. Hemaspaandra, H. Schnoor, On the complexity of elementary modal logics, in: Proceedings of the 25th International Symposium on Theoretical Aspects of Computer Science, Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2008, pp. 349–360.
[20] P. Jeavons, D. Cohen, M. Gyssens, Closure properties of constraints, J. ACM 44 (4) (1997) 527–548.
[21] R. Ladner, The computational complexity of provability in systems of modal propositional logic, SIAM J. Comput. 6 (3) (1977) 467–480.
[22] R. Ladner, J. Reif, The logic of distributed protocols: Preliminary report, in: Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning About Knowledge, Morgan Kaufmann, 1986, pp. 207–222.
[23] D. Lau, Function Algebras on Finite Sets: Basic Course on Many-Valued Logic and Clone Theory, Springer Monogr. Math., Springer-Verlag New York, Inc., Secaucus, 2006.
[24] H. Lewis, Satisfiability problems for propositional calculi, Math. Syst. Theory 13 (1979) 45–53.
[25] C.-J. Liau, Belief, information acquisition, and trust in multi-agent systems – a modal logic formulation, Artificial Intelligence 149 (1) (2003) 31–60.
[26] J. McCarthy, M. Sato, T. Hayashi, S. Igarashi, On the model theory of knowledge, Tech. Rep., Stanford University, 1978.
[27] R. Moore, Reasoning about knowledge and action, Tech. Rep. 191, AI Center, SRI International, 1979.
[28] N. Pippenger, Theories of Computability, Cambridge University Press, Cambridge, 1997.
[29] E. Post, The two-valued iterative systems of mathematical logic, Ann. of Math. Stud. 5 (1941) 1–122.
[30] S. Reith, Generalized satisfiability problems, PhD thesis, Universität Würzburg, 2001.
[31] S. Reith, H. Vollmer, Optimal satisfiability for propositional calculi and constraint satisfaction problems, Inform. and Comput. 186 (1) (2003) 1–19.
[32] S. Reith, K. Wagner, The complexity of problems defined by Boolean circuits, in: Proceedings of Mathematical Foundations of Informatics, 1999, World Science Publishing, 2005.
[33] T. Schaefer, The complexity of satisfiability problems, in: Proceedings 10th Symposium on Theory of Computing, ACM, 1978, pp. 216–226.
[34] M. Schmidt-Schauss, G. Smolka, Attributive concept descriptions with complements, Artificial Intelligence 48 (1) (1991) 1–26.
[35] H. Schnoor, Algebraic techniques for satisfiability problems, PhD thesis, University of Hannover, 2007, http://www.thi.uni-hannover.de/fileadmin/forschung/arbeiten/hschnoor-diss.pdf.
[36] L. Schröder, D. Pattinson, PSPACE bounds for rank-1 modal logics, in: Logic in Computer Science, IEEE Computer Society, 2006, pp. 231–242.
[37] A. Sistla, E. Clarke, The complexity of propositional linear temporal logics, J. ACM 32 (3) (1985) 733–749.
[38] H. Vollmer, Introduction to Circuit Complexity – A Uniform Approach, Texts Theoret. Comput. Sci., Springer-Verlag, Berlin, Heidelberg, 1999.